Universidad Zaragoza
1542

Trabajo Fin de Grado

Planificación de misiones y navegación autónoma de un quadcopter

Autor

Kilian Nicolás Pascual Latorre

Directores

Luis Enrique Montano Gella
José Luis Villarroel Salcedo

Escuela de Ingeniería y Arquitectura
Zaragoza, Febrero de 2016

Escuela de
Ingeniería y Arquitectura
Universidad Zaragoza

DECLARACIÓN DE AUTORÍA Y ORIGINALIDAD

(Este documento debe acompañar al Trabajo Fin de Grado (TFG)/Trabajo Fin de Máster (TFM) cuando sea depositado para su evaluación).

TRABAJOS DE FIN DE GRADO / FIN DE MÁSTER

D./Dª. Kilian Nicolás Pascual Latorre,

con nº de DNI 73010038B en aplicación de lo dispuesto en el art. 14 (Derechos de autor) del Acuerdo de 11 de septiembre de 2014, del Consejo de Gobierno, por el que se aprueba el Reglamento de los TFG y TFM de la Universidad de Zaragoza,

Declaro que el presente Trabajo de Fin de (Grado/Máster) Grado en Ingeniería Electrónica y Automática, (Título del Trabajo) Planificación de misiones y navegación autónoma de un quadcopter,

es de mi autoría y es original, no habiéndose utilizado fuente sin ser citada debidamente.

Zaragoza, 5 de Febrero de 2016

Fdo: Kilian Nicolás Pascual Latorre

A todos aquellos que me han apoyado en este largo camino.

RESUMEN

PLANIFICACIÓN DE MISIONES Y NAVEGACIÓN AUTÓNOMA DE UN QUADCOPTER

Se ha realizado el diseño e implementación de un sistema de navegación autónoma y el control de misión para un UAV, en este caso un quadcopter, que permite al usuario definir *waypoints* que serán seguidos por el quadrotor sin necesitar la intervención de un piloto. Estos *waypoints* se podrán agrupar como un plan de vuelo que se ejecutarán de forma autónoma, efectuando incluso las maniobras de despegue y aterrizaje. El sistema de navegación actúa sobre el autopiloto del quadcopter, que es el encargado de su estabilización.

Se ha desarrollado un generador de trayectorias para controlar el movimiento en los tres ejes del espacio (XYZ), y los ángulos *roll*, *pitch*, *yaw*, que permite controlar la posición del quadrotor en el espacio. También mediante este generador, se controlan las velocidades y aceleraciones máximas realizadas. Se han hecho simulaciones en tres dimensiones del sistema en Matlab para comprobar el correcto funcionamiento del planificador de misiones y la generación de trayectorias, utilizando para ello modelos aproximados de la dinámica del quadrotor.

La implementación de la navegación autónoma y la planificación de misiones se ha realizado en el dsp F2812 de Texas Instruments, utilizando para ello el sistema operativo de tiempo real SYS/BIOS proporcionado en el entorno de desarrollo Code Composer Studio. Dentro de este sistema de tiempo real, se han integrado y calibrado los sensores para conseguir un óptimo funcionamiento del sistema en los entornos en los que operará el quadcopter.

Tras la implementación de todo el hardware y el desarrollo del software, se han efectuado pruebas en el laboratorio para comprobar el funcionamiento del quadrotor, consiguiendo llevar a cabo el despegue y aterrizaje de modo controlado y la estabilización de la orientación.

Índice general

Capítulo 1

Introducción

1.1. Contexto y estado del arte

Un vehículo aéreo no tripulado (Unmanned Aerial Vehicle, UAV) o dron, es una aeronave reutilizable que vuela sin tripulación, capaz de mantener un vuelo controlado mediante la propulsión de éste por hélices o un motor de reacción [14].

Existen drones de diferentes formas que se pueden clasificar en tres grandes grupos: aviones, helicópteros y multirrotores (Figura 1.1). El dron en el que se basa este TFG es un multirrotor de 4 hélices o quadrotor.

(a) Avion Flyox I de Singular Aircraft.

(b) Helicóptero NEO S300 de UMS Skeldar.

(c) Multirrotor comercial.

Figura 1.1: Tipos de drones.

Se empezaron a desarrollar a principios del siglo XX, tras la Primera Guerra Mundial y eran guiados mediante radio control. Con el desarrollo de la tecnología de posicionamiento GPS, es posible en la actualidad guiar mediante una serie de *waypoints* la aeronave utilizando un sistema autopiloto, consiguiendo un vuelo completamente autónomo sin necesidad de un operario. También existe la posibilidad de operar desde una estación remota el UAV utilizando sistemas de visión.

Para la navegación de todos estos sistemas hoy en día se necesitan coordenadas GPS para su localización, por lo que sólo están habilitados para moverse por exteriores ya que en espacios cerrados (interiores) no hay cobertura de los satélites del GPS. Por eso se están empezando

a desarrollar funciones para la navegación por interiores, como son los sistemas basados en acelerómetros y giróscopos o cámaras que controlan la posición del quadrotor mediante el flujo óptico detectado.

La navegación autónoma, ya sea parcial o completa, tiene una gran cantidad de aplicaciones que se están comenzando a desarrollar ampliamente desde hace unos años, gracias a la evolución de la tecnología y a la miniaturización de la electrónica, permitiendo que la investigación de los drones vaya más allá de usos militares, llegando al punto de que hasta un aficionado puede construir su propio dron. Estas aplicaciones van desde las más antiguas, como el control de fronteras o el bombardeo de un objetivo enemigo sin poner en peligro a la tripulación de una aeronave, a las aplicaciones civiles más actuales y en desarrollo como son la prevención, monitorización y extinción de incendios forestales o la búsqueda de personas desaparecidas.

Figura 1.2: Dron de extinción de incendios de NITROFIREX.

Se va a realizar este proyecto en base al interés que promueven estas aplicaciones, aumentando de manera amplia las líneas de investigación futura, ya que para poder probar cualquier uso de los drones, primero es necesario que éste sea capaz de seguir unas ordenes básicas, en este caso coordenadas espaciales que permiten el desplazamiento del dron en el espacio.

Este proyecto se lleva a cabo dentro del grupo de Robótica, Percepción y Tiempo Real de la Universidad de Zaragoza. Éste es uno de los grupos de investigación del Instituto Universitario de Investigación en Ingeniería de Aragón (I3A) y es considerado Grupo de Investigación por el Gobierno de Aragón. Dicho grupo tiene las siguientes líneas de trabajo:

- Localización y Mapeado Simultáneo.
- Visión por Computador y Percepción.
- Comunicaciones y redes ad-hoc.
- Exoesqueletos y procesamiento de bioseñales.
- Aprendizaje: en robótica, optimización Bayesiana, interfaces cerebro-ordenador...
- Robótica Móvil. Planificación y navegación.

El proyecto desarrollado se enmarca dentro de esta última línea de trabajo, en la que el grupo está interesado en pasar de la robótica móvil terrestre a la aérea. Este cambio se ha comenzado a hacer recientemente, por lo que se busca tener una primera experiencia con la planificación y navegación de UAVs que permitan posteriormente desarrollar proyectos y aplicaciones más avanzadas y ambiciosas.

1.2. Objetivos

El objetivo principal es realizar el sistema de navegación autónomo de un quadrotor Flame Wheel F450 de DJI. Para ello es necesario desarrollar un sistema de planificación de trayectorias y movimientos, y un navegador que ejecute el plan elaborado. Para ello se usará el autopiloto Naza M-Lite, que se encargará de la estabilización y el control de los ángulos de la aeronave, permitiendo centrar el trabajo del TFG en la navegación de forma autónoma del quadrotor.

Para lograr este objetivo principal, es necesaria la integración en el quadrotor de una serie de sensores que posibiliten tanto la navegación como el conocimiento propio del estado del dron, así como de una interfaz gráfica mediante la cual interactuar con el quadrotor permitiendo cambiar en tiempo real los planes de vuelo.

Se utilizará el sistema de navegación inercial (INS), mediante el cual se calcula la posición y orientación del dron aplicando las leyes de la inercia. Cabe mencionar que uno de los objetivos del proyecto es dotar al quadrotor de la capacidad de vuelo autónomo no sólo en exteriores (con el uso de GPS) sino también donde no existen medidas GPS, por lo que la utilización de un sistema IMU es necesaria. Para ello se ha adquirido un sensor inercial (IMU) que permite conocer tanto la actitud de la aeronave, como sus aceleraciones. En base a la información dada por el IMU, se irá generando la trayectoria a seguir por el quadrotor para alcanzar el objetivo satisfactoriamente.

Para la medida de las distancias del quadrotor en su plano XY (distancia a una pared u obstáculo) y de la altura a la que se encuentra respecto del suelo, se utilizarán seis sensores ultrasonidos (uno por cada sentido de los tres ejes).

Las comunicaciones se realizarán mediante un módulo de radio XBee que dotarán al dron de comunicaciones inalámbricas.

Todo ello será integrado en el microcontrolador TMS320F2812 de Texas Instruments y programado utilizando el entorno de desarrollo *Code Composer Studio* integrando el sistema operativo de tiempo real proporcionado por éste (*SYS/BIOS*).

1.3. Organización de la memoria

La memoria esta dividida en 6 capítulos:

- **Hardware y software**: en este capítulo se detallan los sensores y el hardware utilizados en el dron. También se analiza el software utilizado para la realización del proyecto.

- **Arquitectura del hardware**: en este apartado se expone el sistema completo que se va a integrar y se realiza un estudio sobre su implementación en el microcontrolador utilizado.
- **Diseño sistema de navegación**: en este apartado se explican los principios del sistema de navegación diseñado, los tipos de trayectorias realizables y también los modelos y controles realizados para la simulación del quadrotor en el ordenador.
- **Implementación**: en este capítulo se explica como se han integrado el hardware y el software, así como la estructuración y funcionalidades de éste. También se realiza la verificación de que el sistema cumple los requisitos de tiempo real.
- **Pruebas realizadas**: en esta sección de detallan algunas de las pruebas realizadas a lo largo del proyecto, siendo estas necesarias para el modelado del sistema, la verificación del software y sensores y la comprobación de distintos controles.
- **Conclusiones**: en este apartado se exponen las conclusiones a las que se han llegado con el trabajo.

Capítulo 2

Hardware y software

En este capítulo se detalla el hardware utilizado en el quadrotor y el software utilizado para la realización del proyecto.

2.1. Hardware

2.1.1. Estructura quadrotor

El chasis del quadrotor es el modelo Flame Wheel F450 de DJI [3] (Figura 2.1). Está formado por cuatro barras en forma de x, unidas mediante dos placas. Las barras son de un material ligero ultra resistente que permite soportar grandes impactos sin romperse la estructura. La dirección de avance frontal está determinada por los dos brazos de color rojo, quedando los blancos hacia atrás. Las placas de sujeción están colocadas en paralelo, quedando hueco entre ellas para colocar la electrónica de navegación. Además, la placa inferior tiene impresas las pistas PCB que conectan la alimentación de entrada a los ESC de cada motor. La estructura tiene un peso de 282 gr y soporta pesos al despegue de hasta 1.6 Kg. Para mayor seguridad en el despegue y aterrizaje, se han colocado 4 patas en la placa inferior que suavizan los aterrizajes.

2.1.2. Naza M-Lite

Para la estabilización del dron, se ha recurrido al autopiloto comercial Naza M-Lite de DJI [4]. Integra un giróscopo de 3 ejes, un acelerómetro de 3 ejes y un barómetro en la misma pieza, necesarios para el control completo del quadrotor. Incorpora también un sensor externo que monitoriza el estado de la batería. Soporta 2 configuraciones distintas de la estructura del quadrotor y hasta 4 de un hexrotor. Mediante una señal PWM se controla la orientación del dron y la potencia de los motores.

Tiene 3 modos de vuelo:

- **Manual Mode**: en este modo, los ángulos roll, pitch y yaw son controlados en velocidad.

Figura 2.1: Flame Wheel F450.

- **Attitude Mode**: este modo de funcionamiento permite controlar los ángulos roll y pitch en posición, obteniendo una mayor estabilidad. Además controla la altitud del dron mediante el barómetro integrado y permite realizar un aterrizaje de emergencia en caso de que la tensión de la batería sea inferior a un umbral establecido por el usuario.

- **GPS Attitude Mode**: este modo de vuelo requiere la adición de un gps externo, que proporciona DJI. Con esta funcionalidad activada, el autopiloto es capaz de mantener al quadrotor en una posición con relativa precisión incluso en condiciones de fuertes rachas de viento o volver al punto desde el que se encendió en caso de pérdida de la señal de radio o batería baja.

2.1.3. TMS320F2812

El F2812 [8] es un procesador digital de señal (DSP) en el que se ha programado el sistema operativo de tiempo real. Tiene una CPU de 32 bits de coma fija con frecuencia de bus de hasta 150 MHz, 16 Kb de memoria RAM y 128 Kb de memoria flash. La arquitectura del procesador es *Hardvard*, con un bus de programa de 22 bits y dos buses de datos de 32 bits. Implementa diversos tipos de periféricos para la comunicación serie, como la SPI, el bus CAN, el McBSP y la SCI, la cual utilizaremos para la comunicación con el IMU y el ordenador. También incorpora un conversor ADC de 16 canales multiplexados en grupos de 8, con una resolución de 12 bits. Implementa dos *Event Managers*, los cuales disponen de un Timer, 3 unidades de captura y 3 unidades de comparación cada uno, con los que se generan las señales PWM y se miden los sensores ultrasonidos.

2.1.4. Sensores ultrasonidos

Para la medida de la altura a la que se encuentra el quadrotor respecto del suelo y la distancia de este al entorno, se utilizan dos tipos de sensores ultrasonidos. Estos sensores, miden la distancia al objeto situado frente a ellos mediante una ráfaga de ondas a 40 KHz y devuelven la distancia medida codificada en tiempo en un pulso, ésto es, el tiempo que el pulso se encuentra en estado alto es proporcional a la distancia medida (Figura 2.2).

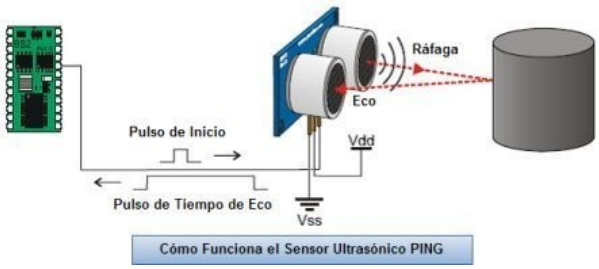

Figura 2.2: Funcionamiento del sensor PING))) [10].

Los 2 sensores ultrasonidos utilizados son:

- PING))): es el sensor utilizado para medir la altura, tiene un rango de 2 cm a 300 cm con una precisión de 1 cm.
- MaxSonar EZ3: se utiliza para medir las distancias laterales , tiene un rango de medida de 20 cm a 654 cm.

2.1.5. Unidad de medición inercial

La estimación de la orientación del quadrotor se realiza mediante el sensor inercial 9DOF Razor IMU [12], el cual integra un acelerómetro, un giróscopo y un magnetómetro, de 3 ejes todos ellos. Incorpora un microcontrolador arduino, el cual mediante una matriz de cosenos directores calcula la actitud del quad. Este sensor nos da la información de los ángulos roll, pitch y yaw del dron, así como las velocidades angulares y las aceleraciones lineales de todos sus ejes a través de la linea SCI.

2.1.6. Sensor de la batería

Para la alimentación del microcontrolador y de los sensores del quad, se ha utilizado el 3DR Power Module [1], que proporciona una tensión de 5.3 V regulada y hasta 2.25 A. También incorpora dos sensores que miden la tensión de la batería y la corriente proporcionada por esta, pudiendo monitorizar el estado de carga en el que se encuentra. Desde el software se mide el voltaje de la batería para actuar de acuerdo a sus niveles y debido a las necesidades de control.

2.1.7. Comunicaciones inalámbricas

La comunicación entre el quadrotor y el ordenador se realiza mediante el módulo XBee Pro S5 [2] (Figura 2.3), un sistema de comunicaciones por radio que utiliza la banda de 868 MHz, el cual tiene un alcance en interiores de hasta 550 m, y en exteriores de hasta 80 Km mediante la inclusión de una antena de alta ganancia. El enlace entre los Xbee con el ordenador y con el quad utiliza la SCI, con lo que la complejidad del sistema de comunicaciones por radio se reduce a una linea serie habitual. La utilización de estos módulos responden a la necesidad de transmitir las ordenes al dron y poder detenerlo en caso de peligro, así como de visualizar y guardar las variables tanto de posición y orientación, como de los controles internos de los ejes mediante la telemetría, para la posterior revisión del correcto funcionamiento del sistema.

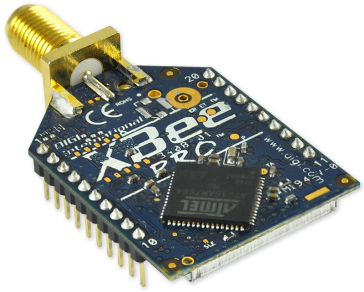

Figura 2.3: Módulo XBee Pro S5.

2.1.8. Multiplexor RF

Se ha utilizado el multiplexor de cuatro canales Pololu 4-Channel RC Servo Multiplexer [11] para seleccionar la entrada de control al autopiloto, pudiendo elegir entre el mando radiocontrol y ser el dron controlado de forma manual, o las señales PWM generadas por el DSP permitiendo controlar el quadrotor de forma autónoma. El multiplexor es capaz de seleccionar entre dos entradas de cuatro canales cada una, ajustando a la salida la entrada seleccionada mediante un quinto canal incorporado en el mando de radiofrecuencia. En este caso se ha utilizado para poder probar los controles en desarrollo, pudiendo controlar de forma híbrida el dron, esto es, controlar a la vez varios canales de forma manual y a través de los controles programados en el microcontrolador.

2.1.9. Equipo radiocontrol

El equipo radiocontrol que se ha adquirido para el control manual del quadrotor consta de un mando transmisor Futaba 6J, con 6 canales y que transmite a 2.4 GHz, y un receptor R2006GS, tambien de 6 canales y a 2.4 GHz [6]. Cinco de los seis canales disponibles son utilizados, siendo el primero para controla el ángulo roll, el segundo para controlar el ángulo pitch, el tercero para el *Throttle* o potencia de los rotores, el cuarto para el ángulo yaw y el quinto para la selección del modo de vuelo, así como de la señal de entrada al autopiloto.

2.2. Software

En esta sección se explica el software utilizado para la programación y realización de pruebas del proyecto:

2.2.1. Matlab

Es un software matemático que ofrece un entorno de desarrollo integrado, el cual se programa en *m*, un lenguaje de programación propio. Su funcionamiento básico se basa en la manipulación de matrices, y sus funcionalidades más destacadas son la implementación y depuración de algoritmos y programas, representación de datos y funciones y el manejo de ficheros. Incluye además una serie de *Toolboxes* o paquetes de herramientas que permiten ampliar sus funcionalidades a campos como el análisis de señales, sector financiero o incluso la programación directa de microcontroladores.

Otra herramienta de Matlab es el GUIDE, un editor de interfaces de usuario con el que se puede realizar una interfaz gráfica mediante la inclusión de elementos predefinidos en la herramienta, simplificando en gran medida la creación de un programa. Genera el código automáticamente y sólo se ha de programar las funciones que se quieran realizar con cada elemento.

2.2.2. Code Composer Studio

Code Composer Studio es un entorno de desarrollo integrado para el desarrollo de aplicaciones en procesadores integrados de Texas Instruments. Incluye un entorno de creación de proyectos, un editor de software, un sistema operativo de tiempo real y varios compiladores y debugger.

2.2.3. SYS/BIOS

SYS/BIOS es un sistema operativo de tiempo real de Texas Instruments para su uso en microcontroladores y microprocesadores [7]. Se usa en aplicaciones que requieren una planificación, sincronización o instrumentación del código en tiempo real. Viene integrado en el software Code Composer Studio de Texas Instruments y tiene licencia de código abierto, por lo que su uso y modificación es libre. Realiza una planificación del procesador basada en prioridades fijas y herencia de prioridad, asegurando el determinismo temporal del software.

Capítulo 3

Arquitectura del sistema

En este capítulo se va a plantear el hardware que se va a integrar en el F2812 y los módulos principales que serán programados en él. En la figura 3.1 se muestra el diagrama de bloques del sistema implementado, además de los puertos utilizados en el microcontrolador.

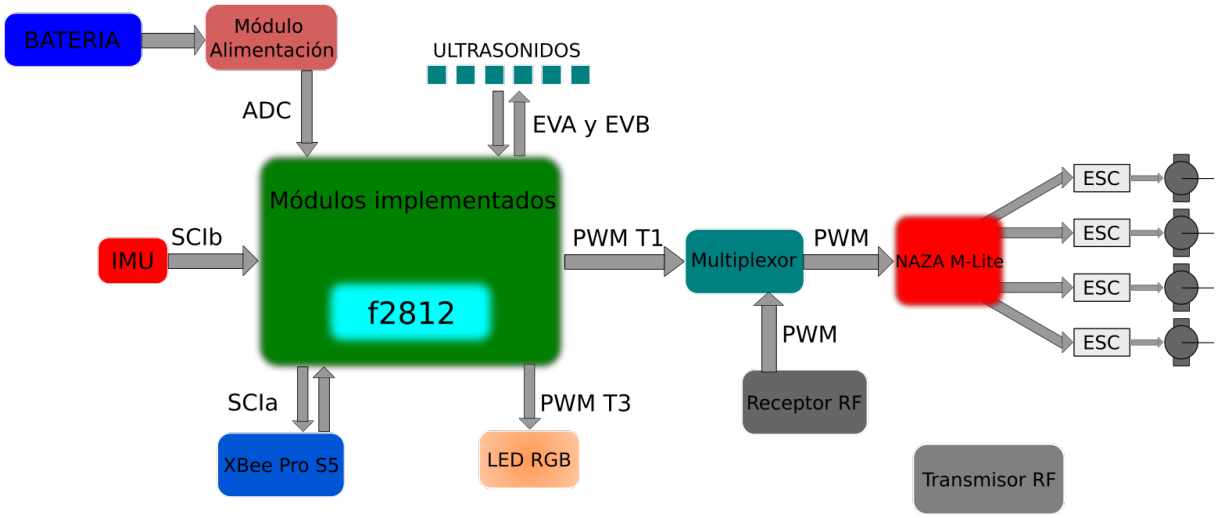

Figura 3.1: Diagrama de bloques del hardware.

3.1. Drivers de los periféricos

Para comunicar los sensores y los actuadores con el dsp se han de realizar una serie de drivers específicos para cada uno de ellos. Para facilitar la lectura de los datos y la realización de acciones se van a utilizar los módulos integrados en el F2812.

3.1.1. Sensores ultrasonidos

Para medir la distancia proporcionada por los sensores ultrasonidos, codificada en tiempo mediante un valor en alto de la onda proporcional a la distancia medida, se dispone del módulo

Event Manager (EVA y EVB). Este módulo incorpora una función para la medida del tiempo transcurrido entre eventos llamada *Input Capture* que permite medir la duración del pulso enviado por los ultrasonidos mediante una interrupción hardware generada cada vez que se detecta un flanco en la señal de entrada. Para ello, se configura la base de tiempos del timer asociado (T2 para el EVA y T4 para el EVB) en función al tiempo que se desea medir. Cada uno de los módulos incorpora 3 unidades de captura, por lo que se dividen los sensores en bloques de 3.

3.1.2. LED RGB y actuación sobre el autopiloto

La comunicación con estos dos periféricos se realiza mediante una señal PWM. En el caso del autopiloto, esta señal contiene la velocidad angular de cada ángulo y la potencia a aplicar en los motores, y en el caso del LED regula la intensidad del color de forma proporcional al valor generado. Para la generación de esta señal, el microcontrolador incorpora en cada módulo *Event Manager* una función que permite hacerlo de forma automática. Esta función es el *Output Compare*, con el que simplemente preescalando el registro *period* fijamos la frecuencia de la señal, modificando la relación del ancho de pulso mediante el registro *compare*.

Se ha asignado al LED RGB el módulo EVB, utilizando los canales PWM7, PWM9 y PWM11 para regular los colores rojo, verde y azul respectivamente.

Para el autopiloto, se utiliza el módulo EVA, del cual se utilizan los cuatro canales disponibles. Éstos son el PWM1, PWM3, PWM5 y el T1PWM que controlan las salidas del roll, pitch, throttle y yaw respectivamente.

3.1.3. Unidad de medición inercial y módulo de comunicaciones inalámbricas

Para la lectura de los datos enviados por el IMU y la comunicación con el módulo XBee se utiliza la línea serie SCI integrada en el microprocesador. Esta línea consta de dos canales (SCIA y SCIB) independientes, los cuales permiten establecer distintas velocidades de transmisión para cada canal y gestionar las interrupciones asociadas a los eventos de llegada y transmisión de datos de forma separada.

El IMU utiliza la línea SCIB y transmite los datos con una tasa de 57600 bps, la máxima que permite el microcontrolador que incorpora. Utiliza solamente la linea de transmisión, ya que ha sido configurado para enviar una trama cada 25 ms sin necesidad de solicitarla desde el procesador. Mediante la interrupción asociada a la llegada de un nuevo carácter, se detecta el inicio de una nueva trama mediante el carácter #, y el final de la trama con el carácter $\backslash n$.

El módulo del XBee utiliza las líneas de transmisión y de recepción de la SCIA, estableciendo el enlace radio entre el F2812 y el ordenador. Las tramas recibidas desde el ordenador se detectan del mismo modo que en el IMU. La gestión de las tramas a enviar se realiza mediante la interrupción asociada a la transmisión de cada carácter, enviando uno nuevo hasta que se detecta el final de línea.

3.1.4. Módulo de alimentación

Se dispone del módulo de conversión analógico/digital (ADC) con el que es posible medir en el dsp la tensión proporcionada por el sensor de la alimentación. Este módulo dispone dos canales con ocho entradas distintas y un *sample & hold* cada uno, utilizando la entrada *ADCINA0* para realizar la conversión.

3.2. Módulos de navegación

Las funciones con las que se desea dotar al quadrotor se pueden agrupar en 4 módulos:

3.2.1. Generación trayectorias

Se desea implementar un generador de trayectorias que permita hacer la planificación del movimiento a realizar entre dos puntos consecutivos de consigna, controlando en todo momento la posición, velocidad y aceleración del dron en cada eje y giro. Desde esta función, se controla el estado del quadrotor en la trayectoria, así como el tipo de movimiento que se está haciendo (avance o retroceso por ejemplo), información que será utilizada por el sistema de navegación. Las referencias generadas serán utilizadas por el control para calcular las acciones a realizar.

3.2.2. Navegación

La navegación del dron se encarga de gestionar los *waypoints* que el usuario desea alcanzar. Establece las consignas para el generador de trayectorias y verifica si éstas han sido alcanzadas. También se encarga de establecer qué controles serán utilizados en cada movimiento y en qué momento, y el cambio entre ellos.

3.2.3. Comunicación

Para establecer una línea de comunicación entre el microcontrolador y el ordenador, es necesaria la implementación de un sistema que permita enviar y decodificar órdenes y datos entre los dos terminales. La transmisión de información desde el ordenador al dsp se realiza mediante comandos predefinidos en éste para poder extraer los datos transmitidos, de modo que se puede desde encender los motores a establecer una nueva consigna de posición y velocidad. Por el contrario, la transmisión que se desea realizar desde el quadrotor hasta el pc son tramas de telemetría en las que se envía toda la información del estado del quadrotor y la proporcionada por los sensores.

3.2.4. Control

El control es la función más importante del software, ya que en él se calculan las acciones a realizar sobre el autopiloto a partir de las referencias calculadas en el generador de trayectorias.

Capítulo 4

Diseño del sistema de navegación

El objetivo principal de este TFG es la realización de un generador de trayectorias que permita el seguimiento de una serie de puntos dispuestos en el espacio. Para ello es necesario controlar los tres ángulos del quadrotor y las tres direcciones del espacio. En este capítulo se va a explicar los pasos seguidos para el desarrollo de los algoritmos de la generación de trayectorias y los controles implementados.

4.1. Ejes y sistemas de referencia del dron

Como se ha explicado en la sección 2.1.1, la dirección del eje X del dron viene determinada por las barras de color rojo, quedando el resto de ejes y giros representados en la figura 4.1:

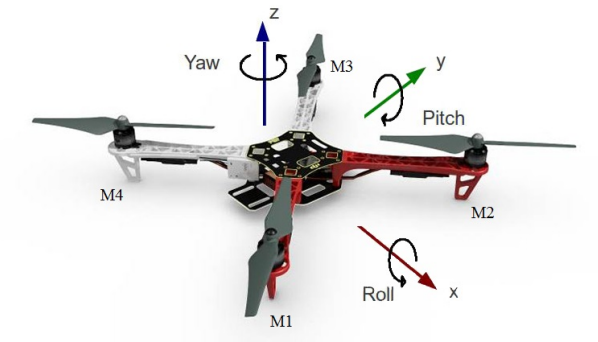

Figura 4.1: Ejes y rotaciones del dron.

Estos ejes se establecerán como base para los tres sistemas de referencia utilizados en la navegación y planificación de misiones. El primero de ellos, es el sistema de referencia R_{IMU}. Este sistema es solidario al chasis del dron, coincidiendo con los ejes mostrados en la figura anterior. Toma el nombre del R_{IMU} debido a que este sensor da la información referida a este sistema.

El segundo sistema de referencia es el denominado R_{Quad}, utilizado para la navegación en el plano XY. Este sistema es necesario para el cálculo del avance y la posición del quadrotor

a través de él. Para ello, se permite el giro del ángulo yaw (rotando con él los ejes XY) y se bloquean los giros roll y pitch (a 0 grados). Este sistema parte de la referencia R_{IMU}, corrigiendo las rotaciones de los ángulos roll y pitch.

Por último, el sistema de referencia R_{Mundo} se establece en el punto en el que se enciende el quadrotor, quedando esas coordenadas XYZ determinadas como $(0, 0, 0)$, y el ángulo yaw orientado al polo norte magnético, pudiéndose reasignar éste en caso de necesidad. Este sistema se mantiene fijo a lo largo de toda la ejecución de la misión, y se utiliza para establecer las consignas globales a las que se desea llegar.

4.2. Principio de funcionamiento

La idea de la que parte el generador de trayectorias es la de realizar un control de posición para alcanzar los *waypoints* de forma precisa, por lo que se decide dividir la trayectoria total en una serie de puntos referencia que permitan realizar un control de la localización del quad sin someterlo a grandes aceleraciones o cambios bruscos. Para ello se realiza el cálculo de las referencias aplicando las ecuaciones del movimiento rectilíneo uniformemente acelerado (mrua). Al aplicar estas ecuaciones, se fuerza al control de posición a seguir un perfil de velocidad y aceleración nunca superiores a las especificadas para su cálculo. De este modo, se realiza un control indirecto de la velocidad y aceleración con las que responde el quadrotor. Las ecuaciones del mrua son:

$$\text{Aceleración} \Rightarrow a(t) = \frac{v_f - v_i}{t_a} = cte \tag{4.1}$$

$$\text{Velocidad} \Rightarrow v(t) = v(0) + at \tag{4.2}$$

$$\text{Posición} \Rightarrow x(t) = x(0) + v(0)t + \frac{1}{2}at^2 \tag{4.3}$$

Para minimizar la memoria utilizada y mejorar la precisión en el cálculo de las referencias, estas ecuaciones se aplican de forma iterada antes de ejecutarse el control del sistema. Ésto quiere decir que la nueva consigna que será ejecutada por el control se calcula a partir de la posición en la que se encuentra en cada instante el quadrotor, siendo la referencia un incremento de posición y velocidad respecto de las actuales. Las referencias calculadas en el instante n serían las siguientes:

$$x_{ref}(n) = x_{act}(n) + v_{act}(n)T + \frac{1}{2}a_t T^2 \tag{4.4}$$

$$v_{ref}(n) = v_{act}(n) + a_t(n)T \tag{4.5}$$

en la que T representa el periodo de ejecución del control y a_t representa la aceleración a

aplicar en el tramo n.

Al realizar la generación de este modo, las referencias calculadas no siguen el perfil de movimiento deseado ya que el sistema realimentado tiene un error de posición en la respuesta, obteniendo una generación muy lenta. Por ello se decide realimentar también las referencias de posición y velocidad calculadas en el instante anterior, quedando las ecuaciones 4.4 y 4.5 de la forma:

$$x_{ref}(n) = x_{act}(n) + v_{act}(n)T + \frac{1}{2}a_t T^2 + (x_{ref}(n-1) - x_{act}(n)) + (v_{ref}(n-1) - v_{act}(n))T$$

$$v_{ref}(n) = v_{act}(n) + a_t(n)T + (v_{ref}(n-1) - v_{act}(n))$$

Operando los términos, obtenemos las ecuaciones 4.6 y 4.7, pudiendo concluir que para poder controlar de forma indirecta la velocidad y aceleración de la trayectoria es necesario generarlas en función de las consignas calculadas en el periodo anterior.

$$x_{ref}(n) = x_{ref}(n-1) + v_{ref}(n-1)T + \frac{1}{2}a_t T^2 \tag{4.6}$$

$$v_{ref}(n) = v_{ref}(n-1) + a_t T \tag{4.7}$$

Esto se puede observar en la siguiente figura, en la que se desea alcanzar una referencia de 50 cm a una velocidad de $10cm/s$. La trayectoria roja representa las referencias de posición obtenidas utilizando la realimentación sin considerar la referencia del periodo anterior y la azul la calculada mediante la realimentación:

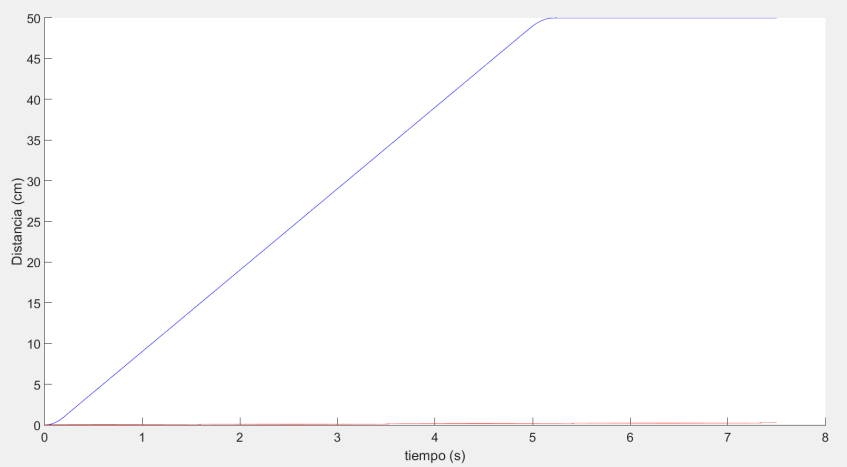

Figura 4.2: Error en la generación de trayectorias de posición. La referencia sin considerar las referencias (roja) se mantiene casi nula.

En este tipo de movimiento, la aceleración constante es la condición básica para que se cumplan estas ecuaciones, por lo que se pueden distinguir tres casos distintos de acuerdo a su valor: si la aceleración es positiva, se produce una aceleración del quadrotor, si ésta es negativa, se produce una deceleración, y si es nula, el dron mantiene una velocidad constante. Atendiendo a estos casos, podemos dividir la trayectoria total en al menos dos tramos (aceleración y deceleración), siendo el caso habitual el movimiento formado por tres tramos. De aquí se pueden extraer los dos tipos básicos de perfiles de velocidad de los que podrá estar formada la trayectoria: el primero de ellos será un perfil de velocidad triangular(Figura 4.3), que estará formado solamente por una aceleración seguida de una deceleración, y el segundo es un perfil de velocidad trapezoidal (Figura 4.4), en el que entre estos periodos de aceleraciones hay un tramo a velocidad constante.

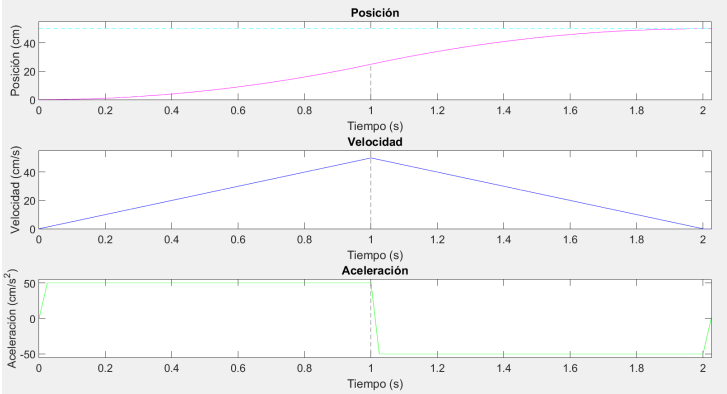

Figura 4.3: Perfil de velocidad triangular.

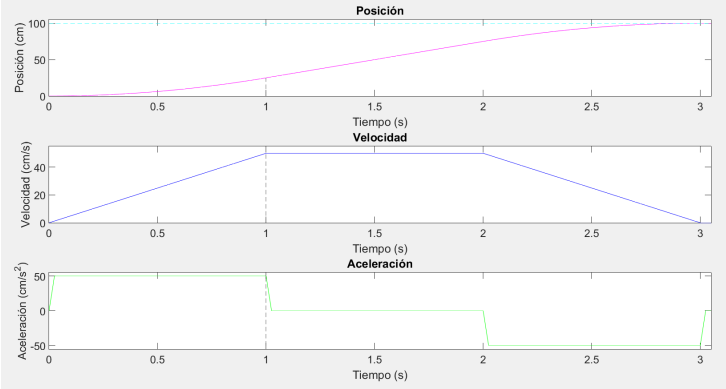

Figura 4.4: Perfil de velocidad trapezoidal.

Se puede observar en la imagen que en ambos casos, la velocidad no supera la deseada y no se producen cambios bruscos en la posición permitiendo alcanzar la consigna establecida de forma suave.

4.3. Tipos de trayectorias realizables

El usuario especifica los movimientos a realizar por el dron mediante uno o varios puntos en el espacio, de la forma XYZ en la referencia del *mundo*, siendo la altura la absoluta sobre el suelo y las distancias en el plano XY relativas al punto de inicio. También especifica el usuario la velocidad deseada para cada movimiento, incluyendo la velocidad de rotación de los ángulos, siendo éstos determinados por los controles de posición.

El control de las 6 variables no es independiente, ya que el movimiento en el eje X está determinado por el ángulo pitch y el del eje Y por el ángulo roll. Para ello se diseñan 6 controladores independientes para las 6 variables teniendo en cuenta las restricciones anteriores. En función del tipo de movimiento que se desee hacer, se pueden distinguir dos formas de controlar los movimientos en el plano XY: la primera de ellas es un movimiento independiente por cada eje (Figura 4.5), descomponiendo la distancia a recorrer en los dos ejes y manteniendo la orientación del ángulo yaw independiente de la trayectoria; y la segunda está formada por un movimiento del eje X y una orientación del ángulo yaw (Figura 4.6), realizando de esta forma un avance frontal siempre. En este caso, el eje Y sólo controla las posibles desviaciones laterales en la trayectoria.

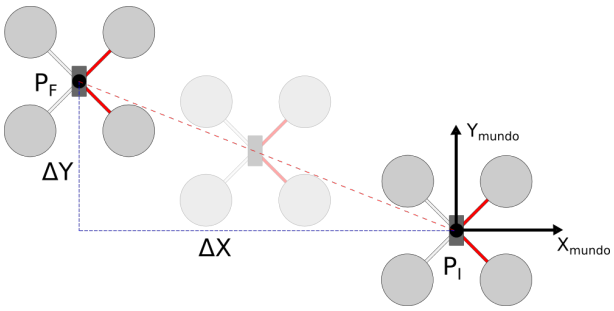

Figura 4.5: Movimiento independiente por cada eje.

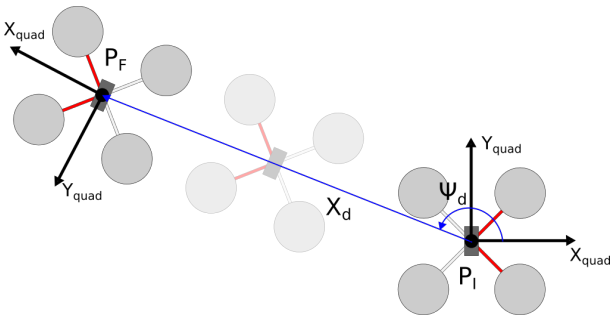

Figura 4.6: Movimiento realizado mediante una rotación en yaw y una traslación en X.

Se han diseñado tres tipos de formas de realizar los movimientos deseados:

- **Trayectoria punto a punto sin orientación**: ésta es la más básica ya que se controlan

independientemente cada eje de movimiento, pudiendo mantener el ángulo yaw orientado de forma fija a lo largo de todo el movimiento. El sistema de referencia utilizado para este movimiento es R_{Mundo}. Se calcula la referencia para cada eje como la variación de posición sobre cada uno de ellos en base al ángulo yaw que se va a mantener (ΔX en el eje X y ΔY en el eje Y en la figura 4.5). Una vez alcanzada la referencia, el dron se para, repitiendo el proceso en caso de que haya una nueva referencia.

- **Trayectoria punto a punto con orientación**: en este tipo de movimiento (Figura 4.6) el sistema de referencia utilizado es R_{Quad}, mediante el cual los movimientos en XY son referenciados al punto de origen de la trayectoria, mientras que las consignas dadas ($X_I Y_I$) y ($X_F Y_F$), inicial y final respectivamente están referenciadas a R_{Mundo}. Antes de comenzar a moverse, se orienta el ángulo yaw hacia la consigna deseada (Ψ_d), quedando determinado por la ecuación 4.9, y una vez en posición se inicia el movimiento sobre el eje X (ec. 4.8) una distancia X_d (en R_{Quad}), controlando con el eje Y los posibles desplazamientos de la trayectoria. Durante todo el trayecto se mantiene el ángulo yaw fijo y correctamente orientado, deteniéndose el quadrotor al alcanzar la referencia. Las consignas de movimiento se calculan aplicando las siguientes ecuaciones:

$$X_d = \sqrt{(X_F - X_I)^2 + (Y_F - Y_I)^2} \tag{4.8}$$

$$\Psi_d = \arctan(\frac{Y_F - Y_I}{X_F - X_I}) \tag{4.9}$$

- **Trayectoria con interpolación entre consignas**: este tipo se aplica en casos en los que hay al menos tres consignas consecutivas, consistiendo el movimiento en pasar cerca de las referencias intermedias deteniéndose sólo en la consigna final, consiguiendo así una trayectoria más rápida ya que no hay que pararse en cada punto. Es el movimiento más complejo de todos, ya que requiere reorientar el ángulo yaw durante el avance a lo largo del eje X, quedando de nuevo libre el eje Y para realizar correcciones laterales. En este tipo de movimiento se utiliza principalmente un control de velocidad para el eje X, ya que para reducir las no linealidades del movimiento se requiere mantener una velocidad lo más constante posible durante la rotación del ángulo yaw. Una vez cerca de la consigna final, se cambia al control de posición en el eje X para detener el quadrotor lo más preciso posible respecto a la referencia.

4.4. Modelo de quadrotor para simulación de trayectorias

Para verificar el funcionamiento del generador de trayectorias y el comportamiento del quadrotor con las referencias calculadas, se ha modelado el quad a partir de las ecuaciones de [5]. Éstas han sido simplificadas a un modelo lineal para facilitar su aplicación, ya que el autopiloto

se encarga de controlar completamente el quadrotor.

$$\text{Eje X} \Rightarrow \ddot{x} = -\frac{b_x}{m}\dot{x} + g\tan\theta \tag{4.10}$$

$$\text{Eje Y} \Rightarrow \ddot{y} = -\frac{b_y}{m}\dot{y} + g\tan\phi \tag{4.11}$$

$$\text{Eje Z} \Rightarrow \ddot{z} = -\frac{b_z}{m}\dot{z} + \frac{1}{m}T - g \tag{4.12}$$

$$\text{Ángulo Yaw} \Rightarrow \dot{\psi} = u_\psi \tag{4.13}$$

Las variables del modelo simplificado son las siguientes:

- x, $\dot{x}$, $\ddot{x}$: posición, velocidad lineal y aceleración lineal en la dirección de avance del dron.
- y, $\dot{y}$, $\ddot{y}$: posición, velocidad lineal y aceleración lineal en la dirección lateral del quad.
- z, $\dot{z}$, $\ddot{z}$: posición, velocidad y aceleración del eje vertical del dron.
- ϕ: ángulo roll del quadrotor.
- θ: ángulo pitch del dron.
- ψ, $\dot{\psi}$: posición y velocidad angular del ángulo yaw.

Las constantes físicas del sistema se presentan a continuación, estando su valor detallado en el apéndice (A):

- b_x, b_y, b_z: fricción viscosa estimada para cada eje de control.
- m: masa del quadrotor.
- g: gravedad terrestre.
- T: fuerza total ejercida por las hélices en el eje Z.
- u_ψ: acción del control del ángulo yaw, físicamente es la velocidad angular aplicada al autopiloto.

Debido a que los ángulos roll y pitch los controla el autopiloto en posición, no se considera un modelo matemático para su simulación.

También se ha modelado la conversión de la fuerza de cada hélice a su equivalente en velocidad angular. Ésto es necesario para hacer la posterior conversión de la fuerza de cada una de ellas al PWM a aplicar como entrada *Throttle* al autopiloto:

$$\omega_{helice} = \sqrt{\frac{T}{4C_t\rho\pi R^4}} \tag{4.14}$$

en las que ω_{helice} corresponde a la velocidad angular de cada hélice, C_t es el coeficiente de sustentación de las hélices, ρ es la densidad del aire y R el radio de cada hélice.

4.5. Diseño de controles

Se ha optado por la realización de un control de posición para los ejes XYZ y para la orientación roll, pitch, yaw (RPY), además de un control de velocidad para los ejes XY. En este apartado se van a explicar los esquemas de control que se han incorporado finalmente al montaje real. Las variables que se van a controlar son la posición en el eje Z o altura, y los ángulos roll, pitch y yaw. Todos ellos se realizan en espacio de estados dada su mayor precisión al realimentar todas las variables de estado, obteniendo un mejor control al disponer de mayor información del sistema.

4.5.1. Control de posición de la altura

Se ha optado por implementar un control por realimentación del estado y un controlador integral serie sobre la variable a controlar. Este regulador permite conseguir un error de posición nulo en régimen permanente, además de minimizar los errores provocados por la incertidumbre del modelo asociados a la linealización y simplificación de éste [9]. Para solucionar los efectos de la gravedad, se utiliza una prealimentación de perturbación constante, que corresponde a la aceleración de la gravedad. De este modo, se puede entender que el control realiza pequeñas aceleraciones alrededor de la fuerza equilibrio.

Las leyes de control L_S y L_R se han calculado aplicando el método de *Ackermann* que optimiza la ubicación de polos del sistema en función del tiempo de respuesta deseado. Este método está implementado en la función *acker* de Matlab, que se ha utilizado para su cálculo. El tiempo de respuesta se ha ajustado en las pruebas posteriores, seleccionando el que mejor comportamiento mostraba. Se ha elegido finalmente el controlador con tiempo de respuesta de 0.5 segundos teniendo en cuenta que las acciones (aceleraciones) que es capaz de hacer están limitadas a las posibilidades físicas del sistema.

El esquema de control utilizado es el representado en la figura 4.7.

- $x_s(k)$: es el vector de estado del modelo, lo componen las variables z o altura y $\dot{z}$ o velocidad de ascenso.
- $y(k)$: es la variable de salida del sistema, que es la altura.
- $r(k)$: es la consigna de altura que se desea alcanzar.
- $e(k)$: es el error de posición que hay entre la referencia y la altura realimentada.

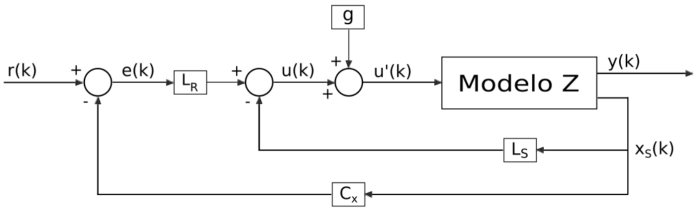

Figura 4.7: Estructura de control utilizada en el eje Z.

- $u(k)$: es la acción calculada por el regulador.
- $u'(k)$: es la acción corregida frente a la perturbación de la gravedad.

Las matrices de control L_S y L_R finalmente programadas son:

$$L_s = \begin{pmatrix} 474{,}3686 & 39{,}6742 \end{pmatrix} \tag{4.15}$$

$$L_r = \begin{pmatrix} 80{,}5054 \end{pmatrix} \tag{4.16}$$

La matriz C_X selecciona la altura del vector de estado $x(k)$ para su realimentación y calcular el error con la referencia:

$$C_X = \begin{pmatrix} 1 & 0 \end{pmatrix} \tag{4.17}$$

En la siguiente imagen se puede observar la respuesta del control realizado para una altura deseada de 100 cm y una velocidad de referencia de 25 cm/s:

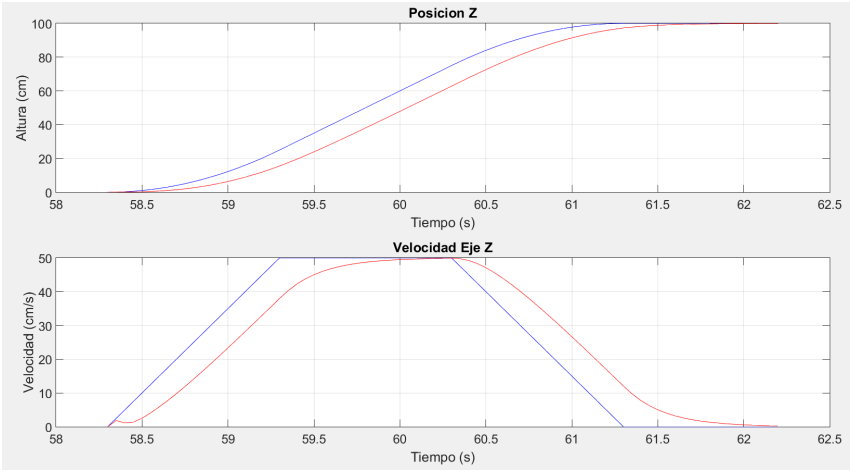

Figura 4.8: Respuesta del control de altura para una consigna de 100 cm. La referencia es la línea azul y la roja la posición alcanzada.

4.5.2. Control de posición angular

Para la realización de las pruebas sobre el quadrotor, se ha tenido que desarrollar un control de posición que permita mantener estable el dron. Para ello, el autopiloto dispone de un control de velocidad para estos ángulos, sobre los que actúa el control de posición. Por eso, la salida del control es la velocidad angular a aplicar directamente sobre el autopiloto.

Ángulos roll y pitch

Para los ángulos roll y pitch se han probado dos reguladores distintos. El primero de ellos es un integrador para eliminar los errores asociados al modelado del sistema, pero tras diversas pruebas sobre el quadrotor, se ha desestimado ya que se inestabiliza y oscila alrededor de la referencia por efecto de la discretización en el tiempo. El segundo regulador consiste en una prealimentación de consigna, obteniendo mejores resultados durante las pruebas y sin tener apenas error en régimen permanente.

Las leyes de control L_S y L_C se han calculado mediante la función *place* de Matlab, que implementa un método optimizado de ubicación de polos para sistemas con una o múltiples variables a controlar. El tiempo de respuesta que mejor ajusta su comportamiento es de $t_r = 0{,}5$ segundos, el mínimo que el hardware permite aplicar debido a sus limitaciones que cumpla con el teorema de Nyquist-Shannon.

El esquema de control utilizado es el siguiente:

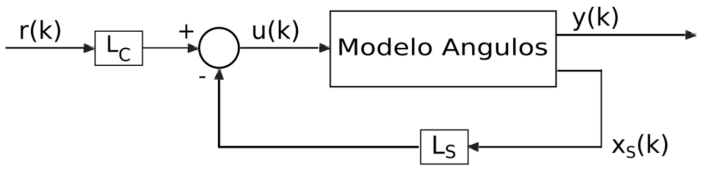

Figura 4.9: Estructura de control utilizada en el control de los ángulos roll y pitch.

- $x_s(k)$: es el vector de estado del modelo, lo componen las variables ϕ y θ.
- $y(k)$: son la salida del sistema, en este caso corresponde también con las variables ϕ y θ.
- $r(k)$: son las referencias angulares roll y pitch deseadas.
- $u(k)$: son las acciones calculadas por el regulador.

Las matrices de control L_S y L_C finalmente programadas son:

$$L_s = \begin{pmatrix} 7{,}5623 & 0 \\ 0 & 7{,}5623 \end{pmatrix} \tag{4.18}$$

$$L_c = \begin{pmatrix} 7{,}5623 & 0 \\ 0 & 7{,}5623 \end{pmatrix} \tag{4.19}$$

En la siguiente imagen se puede observar la respuesta del control realizado para una consigna angular de 10 grados y una velocidad de referencia de 0.5 rad/s para los ángulos roll (Figura 4.10) y pitch (Figura 4.11):

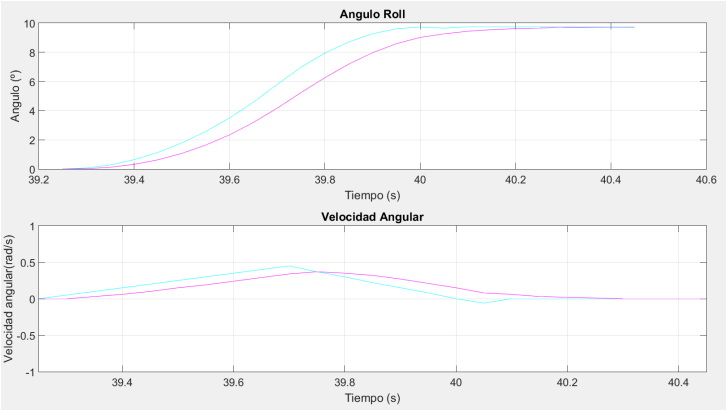

Figura 4.10: Respuesta del control del ángulo roll. La referencia es la línea azul y la roja la posición alcanzada.

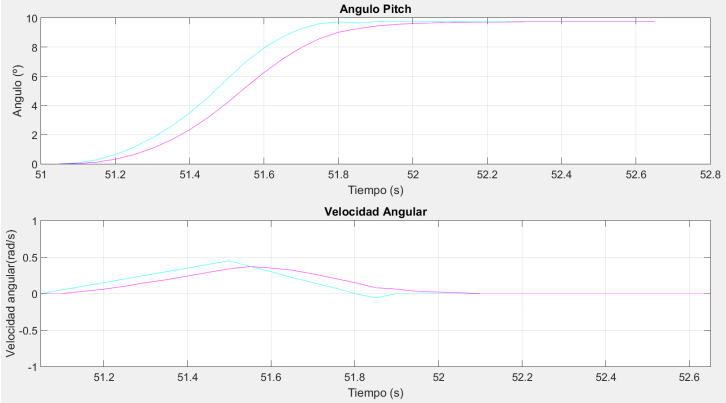

Figura 4.11: Respuesta del control del ángulo pitch. La referencia es la línea azul y la roja la posición alcanzada.

Ángulo yaw

En el ángulo yaw, se han puesto a prueba los mismos reguladores que en los ángulos roll y pitch. En este caso el esquema de prealimentación de consigna tiene una buena respuesta, pero el error es notable frente al sistema con integrador, que consigue obtener una muy buena precisión en el régimen permanente respecto de la consigna sin inestabilizarse. Esto es debido a que este ángulo es más estable que los anteriores. Por ello se utiliza finalmente un esquema de realimentación del estado y un integrador serie del error de la variable controlada.

Las leyes de control L_S y L_R se han calculado mediante la función *acker* de Matlab. El tiempo de respuesta que mejor ajusta su comportamiento es de $t_r = 0{,}5$ segundos, teniendo las mismas restricciones hardware que los ángulos roll y pitch.

El esquema de control utilizado es el siguiente:

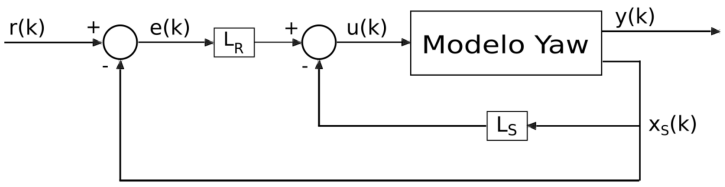

Figura 4.12: Estructura de control utilizada en el control del ángulo Yaw.

- $x_s(k)$: es la variable de estado del modelo ψ.
- $y(k)$: es la salida del sistema, coincidente con el ángulo yaw.
- $r(k)$: es la referencia angular yaw deseada.
- $e(k)$: es el error de posición que hay entre la referencia y el ángulo realimentado.
- $u(k)$: es la acción calculada por el regulador.

Las matrices de control L_S y L_R finalmente programadas son:

$$L_r = \begin{pmatrix} 25{,}702 \end{pmatrix}$$
$$L_r = \begin{pmatrix} 6{,}8589 \end{pmatrix}$$

En la siguiente imagen se puede observar la respuesta del control realizado para una consigna angular de 90 grados, con una velocidad angular de referencia de 1 rad/s:

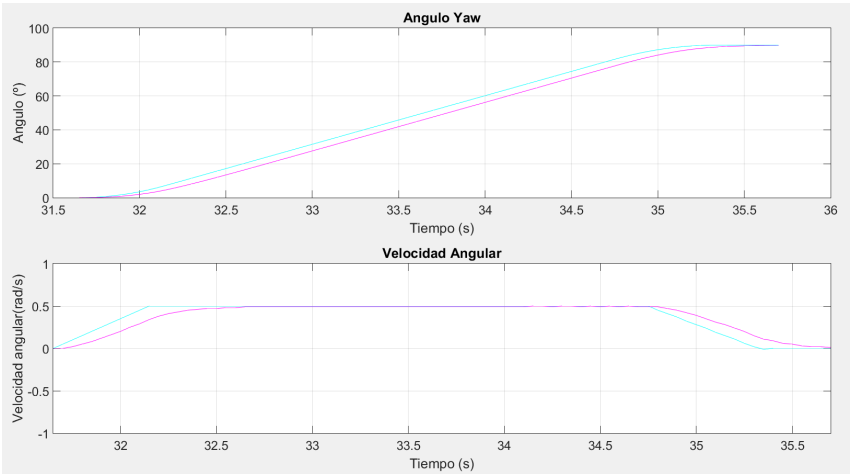

Figura 4.13: Respuesta del control del ángulo yaw. La referencia es la línea azul y la roja la posición alcanzada.

4.6. Simulación del sistema realizado

Para comprobar el correcto funcionamiento del software desarrollado, se ha realizado un entorno de simulación en Matlab que representa en tres dimensiones el comportamiento del quadrotor, pudiendo verificar visualmente el correcto seguimiento de las trayectorias y la correcta ejecución de la misión planificada. Mediante estas simulaciones, se comprueba los desplazamientos y las velocidades de cada eje a lo largo de las trayectorias generadas, permitiendo ver el comportamiento ante perturbaciones como ráfagas de aire u otras perturbaciones.

En la siguiente figura, se muestra el desplazamiento del quadrotor en el sistema de referencia R_{Mundo} en el que el quadrotor tiene que realizar la maniobra de despegue, un cuadrado de 2 metros de lado con un movimiento de punto a punto con orientación a altura constante y un aterrizaje:

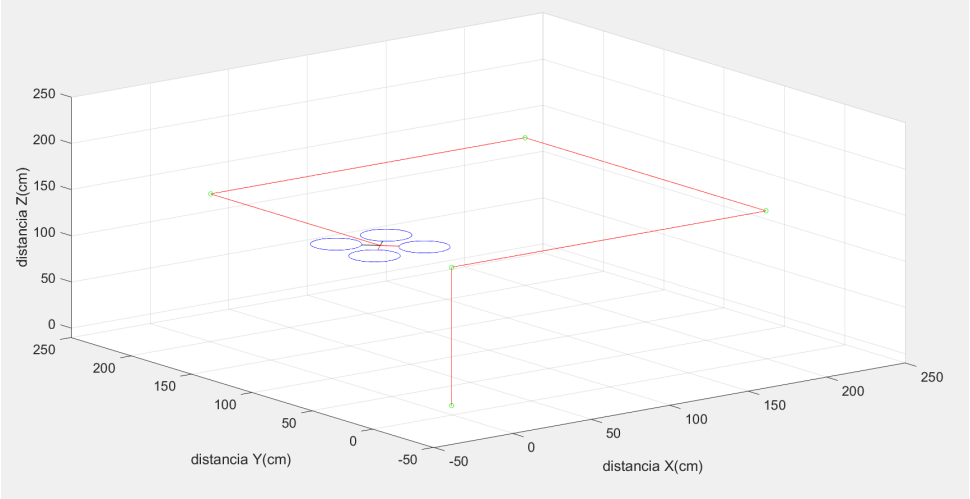

Figura 4.14: Simulación en 3D de una misión.

Capítulo 5

Implementación

El software se ha implementado en un sistema operativo de tiempo real dados los requerimientos temporales exigidos por el quadrotor para su correcto funcionamiento. Estos sistemas se caracterizan por el determinismo temporal en su ejecución, lo que permite optimizar el funcionamiento del control del quadrotor y la planificación de las trayectorias.

5.1. Diseño del software

Las actividades que se han planificado y se van a implementar en el sistema de tiempo real son: medida de la tensión de la batería, recepción de comandos enviados desde el ordenador, transmisión de telemetría al ordenador, gestión del LED RGB, medida de la altura con los sensores ultrasonidos, control del quadrotor, generación de trayectorias y medida y tratamiento de datos del IMU.

Todas estas actividades se agrupan en procesos o tareas que tengan funciones comunes y tiempos de ejecución similares. El núcleo del sistema operativo se encarga de entrelazar las tareas y de gestionar su activación.

Dependiendo del período de activación se pueden distinguir dos tipos de tareas:

- **Tareas periódicas**: se activan en instantes de tiempo regulares, separados por un periodo determinado.
- **Tareas esporádicas**: se activan de forma irregular sin un periodo de separación determinado, mediante eventos externos ya sean interrupciones hardware o software.

En función del período de activación y el plazo de respuesta, se establecen las prioridades de ejecución entre las tareas. De esta forma, se establece el orden en el que se ejecutarán las tareas y el momento en el que se activan para garantizar los requisitos temporales de cada una de ellas [13].

El flujo de información entre las distintas tareas y los servidores está representado en el siguiente diagrama:

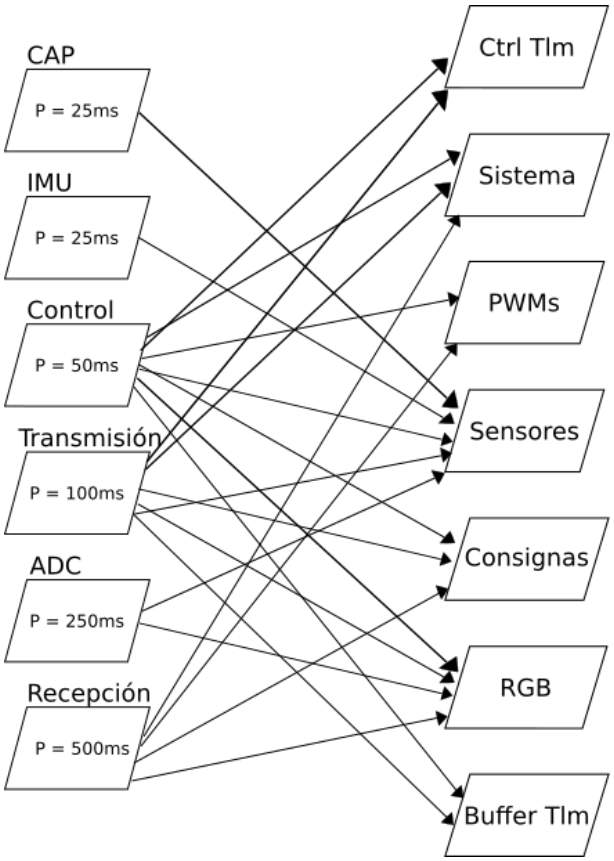

Figura 5.1: Diagrama de tareas y servidores.

5.1.1. Tareas

Las actividades previamente planteadas, se han agrupado en seis tareas ordenadas de menor a mayor prioridad:

- **Recepción de comandos enviados desde el ordenador**: es una tarea esporádica, iniciada cada vez que llega una trama completa desde el ordenador por la línea SCIa a través del módulo XBee.

  Decodifica y ejecuta las órdenes o consignas enviadas, guardando las modificaciones efectuadas en las variables en sus respectivos servidores.

- **Medida de la batería con el ADC**: esta tarea se encarga de medir la tensión de la batería, proporcionada por el módulo de alimentación y de supervisar su estado.

  Dado que la batería reduce su tensión en la descarga lentamente, se ha optado por realizar la conversión cada 0.25 segundos mediante un reloj. Éste inicia la conversión, activando la tarea mediante una interrupción hardware que activa el semáforo. Una vez calculada la tensión, ésta se guarda en el servidor de sensores.

- **Transmisión de telemetría y control del LED RGB**: en esta tarea se han agrupado

dos actividades relacionadas con la comunicación del estado del quadrotor. Mediante la telemetría se envía al ordenador el valor de las variables internas y de control y con el LED se tiene un *feedback* visual del estado del quadrotor, como el estado de la conexión con el pc o el nivel de la batería.

Por limitaciones en la transmisión de datos del XBee, se realiza esta tarea cada 0.1 segundos, tiempo que permite el envío correcto de las tramas sin errores.

- **Control del quadrotor**: es una tarea periódica que se ejecuta cada 50 ms. Es la más compleja de todas las tareas, puesto que se encarga del control de misión, planificación de trayectorias, generación de referencias y el control del quadrotor. También gestiona el encendido de los motores y la reasignación de las constantes de cada control.
- **Medida del IMU**: esta tarea es periódica, iniciada al llegar una trama completa desde el IMU por la línea SCIb cada 25 ms. Cuando se detecta la trama completa desde la interrupción hardware, se inicia la tarea activando el semáforo. En ella se calibran las medidas y se guardan en el servidor de sensores.
- **Medida de la altura**: un reloj cada 25 ms inicia la medida de la altura mediante un sensor ultrasonidos, iniciando también un *timer* que a los 21 ms lanza el semáforo correspondiente a la tarea. Ésta, mide la altura detectada y la almacena en el servidor de sensores.

5.1.2. Servidores

La información compartida entre distintas tareas se ha agrupado en ocho servidores:

- **Sensores**: este servidor almacena todas las variables proporcionadas por los sensores. En él se guardan por separado los datos del IMU, los sensores ultrasonidos y la tensión de la batería.
- **Sistema**: este servidor guarda el estado del quadrotor, de las conexiones y los controles. Esta información es necesaria tanto para la activación de los controles como para la visualización en el ordenador del movimiento del quad.
- **PWMs**: este servidor almacena los PWMs aplicados a cada entrada al autopiloto.
- **Consignas**: las consignas globales a las que se desea llegar se almacenan en este servidor. Se utilizan para la generación de las trayectorias hasta ellas y para detectar que se han alcanzado.
- **Control telemetría**: en este servidor se guardan las variables más importantes de los controles implementados para enviarlas al pc mediante la tarea de telemetría. Sólo permite almacenar un dato por variable debido a las restricciones del uso del módulo XBee.
- **RGB**: en este servidor se guarda el color a mostrar por el LED RGB y la información relevante para su gestión desde la tarea de control del LED.

- **Buffer telemetría**: este servidor almacena todas las variables de control del controlador seleccionado desde el pc. A diferencia del servidor de control, éste almacena todas las variables hasta que son enviadas al ordenador.

5.2. Software implementado

Para el manejo de los distintos sensores y periféricos, se han diseñado una serie de drivers explicados a continuación:

5.2.1. Sensor inercial

El IMU ha sido reprogramado para transmitir además de la orientación del dron, la aceleración lineal y la velocidad angular de éste, variables necesarias para el correcto control de los ángulos y la posición.

Cada 25 ms envía una trama de caracteres a través de la línea SCIb con la misma estructura, por lo que se ha creado una función que decodifica el mensaje, convierte y guarda en sus respectivas variables cada sección de caracteres.

Se desean obtener dos tipos de medida de la aceleración, la referente al eje local al quadrotor y la que está referenciada a los ejes fijados en el inicio. Las aceleraciones referidas al quadrotor son las que manteniendo el ángulo yaw solidario a éste, corrigen la rotación sobre los ejes roll y pitch, obteniendo las aceleraciones lineales en el plano XY real. Estas aceleraciones se utilizan para calcular la posición a lo largo de una trayectoria desde el punto de inicio de ésta. Las aceleraciones referenciadas sobre los ejes iniciales, aplican una nueva corrección sobre las anteriores, corrigiendo la desviación provocada por el ángulo yaw. Estas aceleraciones se utilizan para calcular la posición del dron en el plano absoluto XY.

Una vez calculadas las aceleraciones, se corrigen las desviaciones de los ángulos roll y pitch debidas al montaje, teniendo de esta forma una mayor precisión en el control de la orientación.

5.2.2. Sensores ultrasonidos

Cada vez que se desea realizar una medida, se envía un pulso de 25 μs al sensor para comenzarla. Inmediatamente después de ésto, se inicia un *timer* con un periodo de 21 ms que se encargará de lanzar el semáforo de la tarea encargada de su gestión.

A continuación, se cambia el pin a la función *Input Capture*, que permite medir el tiempo transcurrido entre dos eventos. Cada vez que detecta un flanco en la señal de entrada, almacena el valor temporal en el que se ha producido, resultando el tiempo de la medida la diferencia entre el segundo flanco y el primero. La relación entre el tiempo medido y la distancia detectada está determinada por la ecuación:

$$\text{Distancia medida } (cm) = (\text{ticks medidos}) \cdot 0{,}853 \frac{\mu\text{s}}{\text{tick}} \cdot \frac{1}{1000} \frac{\text{ms}}{\mu\text{s}} \cdot 34{,}32 \frac{\text{cm}}{\text{ms}} \cdot \frac{1}{2} \tag{5.1}$$

Una vez obtenida la medida, se corrige la desviación provocada por la rotación de los ángulos roll y pitch y se le resta la distancia de corrección calibrada en las pruebas para obtener la medida de altura real.

5.2.3. XBee

El módulo de comunicaciones inalámbricas utiliza la línea SCIa para la comunicación bidireccional entre el quad y el ordenador. El envío de información desde el quadrotor a la computadora es principalmente de telemetría referente a la posición y orientación de éste, así como de las variables de control seleccionadas desde el ordenador. Esta información se utiliza para visualizar en tiempo real el funcionamiento del dron y para su posterior análisis y posible detección de errores en el funcionamiento. Los datos de telemetría han sido organizados en tres clases:

- **Telemetría básica**: esta trama contiene información sobre el estado del quadrotor (motores encendidos, si está en tierra o volando o si está activo el control por ejemplo), sobre la tensión de la batería y el tiempo en el que ha sido enviada (referido a la primera conexión entre el ordenador y el microcontrolador).

- **Telemetría general**: esta trama contiene información sobre las posiciones y referencias de los ángulos, la posición y la referencia de altura, la velocidad de ascenso y las acciones ejecutadas por los controles. Debido a las limitaciones de transmisión, sólo se envían los datos correspondientes a la última ejecución del control antes de ejecutarse la tarea de envío, teniendo un fin meramente informativo. Se selecciona desde el pc como trama de tipo 1.

- **Telemetría específica**: esta trama de telemetría envía toda la información guardada referente al control seleccionado desde el pc (tipos 2 al 5). Se utiliza para ver más en profundidad el comportamiento de un control específico.

Los comandos enviados desde el pc al quad son principalmente de control sobre las acciones que realiza, como puede ser el activado del sistema y el encendido de los motores, aunque también integra la posibilidad del cambio de las constantes de los controles y de otras variables de calibración. Estos comandos están previamente definidos en el dsp, de forma que cada vez que llega una trama completa, la tarea llama a una función que los decodifica, y realiza la actividad asociada.

Se ha implementado como opción de seguridad el envío de la cadena de caracteres *Check* cada cinco segundos al ordenador. Al terminar de enviarse, se activa un timer que en caso de que en el plazo de 300 ms no se reciba la cadena ACK, reenvía la cadena de seguridad. Si a los tres intentos, no se recibe ninguna respuesta, se activa un flag que desde la tarea de envío de telemetría establece el sistema como desconectado y activa una luz de emergencia en el LED RGB. Del mismo modo, si en el pc no se detecta la cadena de seguridad cada 7 segundos, se

indica en la interfaz que hay problemas en la comunicación. Tras producirse tres fallos seguidos, se muestra un mensaje de error y se desconecta el enlace radio.

Para aumentar la seguridad en la transmisión de órdenes desde el ordenador al quadrotor, se puede activar la opción de que tras cada orden enviada, se espere un ACK del quad. Si éste no es detectado, se vuelve a enviar la misma orden. Si tras tres intentos no se recibe confirmación, se muestra un error por pantalla y se desconecta la comunicación. Por el contrario, para el envío de la telemetría desde el dron al ordenador, no se estima la necesidad de verificar la correcta transmisión, puesto que son datos meramente informativos y carecen de importancia para el correcto funcionamiento.

5.2.4. Módulo alimentación

El módulo de alimentación utilizado, provee de una salida que proporciona una tensión continua proporcional al nivel de carga al que se encuentra la batería. Esta tensión es medida por el conversor ADC que integra el microcontrolador. Una vez realizada la conversión, una interrupción hardware lanza el semáforo correspondiente a la tarea encargada de su lectura. Esta tarea convierte la tensión medida a la tensión real mediante la siguiente relación:

$$\text{Tensión batería} = (\text{Valor conversor}) \cdot \frac{3}{4095} \cdot 10 \tag{5.2}$$

La fracción $\frac{3}{4095}$ viene dada por la documentación del conversor, mientras que la constante *10* es la relación proporcional entre la tensión real de la batería y la proporcionada por el sensor.

Para la correcta medida de la batería, se ha soldado en la entrada del conversor un filtro RC paso bajo para eliminar el ruido en la señal, además de corregir por programa el error constante en la medida.

5.2.5. Autopiloto

Las órdenes de potencia de los motores y velocidad angular del quadrotor se transmiten mediante una señal PWM al autopitolo. Para la creación de ésta, se ha tomado como referencia las características de la señal transmitida por el control de radiofrecuencia, ya que durante las pruebas y para la utilización de la navegación autónoma se han de emplear las dos señales a la vez. Por ello, se ha de generar una onda PWM de 73 Hz de frecuencia, con un tiempo en alto de la señal que oscila entre 1 y 2 ms.

Desde la tarea de control, se calculan las acciones a aplicar al autopiloto y se codifican en la señal PWM.

5.2.6. LED RGB

El LED RGB, del mismo modo que el autopiloto, se controla mediante una señal PWM. Para cada color primario, se necesita una señal independiente, por lo que se usan tres canales distintos. La frecuencia del PWM se ha elegido de 1 KHz, pudiendo controlar la intensidad

de brillo de cada color variando el valor del 0 al 100 %. De este modo también es posible la visualización de cualquier color mezclando los tres disponibles, regulando la intensidad de cada uno de ellos. Para facilitar ésto, se han predefinido varias funciones que establecen los colores más utilizados: rojo, verde, azul, amarillo, cían, magenta y blanco.

5.3. Análisis de tiempo real

La planificación de las tareas se ha realizado asignando las mayores prioridades a las tareas más frecuentes, y el acceso a los recursos compartidos entre ellas se hace mediante un mutex con protocolo de herencia de prioridad, que permite controlar los casos en los que una tarea de menor prioridad tiene acceso a un recurso compartido que va a utilizar una tarea de mayor prioridad que se está ejecutando. Para determinar si la planificación de las tareas cumple con los plazos fijados, se realiza un análisis de los plazos de respuesta de cada tarea. Para ello se han medido los tiempos de cómputo de cada una de ellas, así como el tiempo de acceso a los servidores (Tabla 5.1):

Tarea	Prioridad	C(μs)	D(ms)	P(ms)	$B_{HP}(\mu s)$
Captura	6	111	25	25	47
IMU	5	981	25	25	47
Control	4	1268	50	50	152
Envío	3	8146	100	100	91
ADC	2	157	250	250	43
Recepción	1	352	500	500	-

Cuadro 5.1: Datos utilizados para la verificación del cumplimiento de plazos de las tareas.

Las variables involucradas son:

- C: es el tiempo de cómputo de cada tarea, medido en μs.
- D: es el plazo de respuesta de la tarea, medido en ms.
- P: es el periodo de ejecución de las tareas, medido en ms.
- B_{HP}: es el tiempo de bloque del peor caso de cada tarea debido a la inversión de prioridad. Está medido en μs.

El factor de utilización del procesador (U) da una idea de la carga de trabajo que lleva, quedando patente el bajo coste computacional del sistema implementado. Está determinado por la fórmula:

$$U = \sum_{j=1}^{n} \frac{C_j}{P_j} = 15{,}18\,\% \tag{5.3}$$

A pesar de tener una carga muy baja, esto no es condición suficiente para poder afirmar que se cumplen los plazos de respuesta de cada una de las tareas.

Aplicando la siguiente fórmula a cada tarea, se comprueba que todas las tareas cumplen los plazos de respuesta:

$$W(D_i) = \sum_{j=1}^{i} \lceil \frac{D_i}{P_j} \rceil C_j + B_i < D_i \tag{5.4}$$

El trabajo solicitado al procesador por cada una de ellas resulta:

$$W(Captura) = 158\mu s < 25ms.$$

$$W(IMU) = 1{,}139ms < 25ms.$$

$$W(Control) = 3{,}604ms < 50ms.$$

$$W(Envio) = 15{,}141ms < 100ms.$$

$$W(ADC) = 41{,}898ms < 250ms.$$

$$W(Recepcion) = 75{,}916ms < 500ms.$$

cumpliendo en todas ellas que el trabajo solicitado al procesador es menor que el plazo de respuesta fijado.

Capítulo 6

Pruebas reales

- **Ajuste del autopiloto**:

  Para la realización de un modelo aproximado de las acciones producidas por el autopiloto sobre el quadrotor, se han llevado a cabo una serie de pruebas que permitan modelar la fuerza ejercida por las hélices en función del PWM aplicado a la entrada *throttle* del autopiloto, y la relación entre las velocidades angulares reales y las entradas PWM sobre los ángulos roll, pitch, yaw. Este paso es necesario para poder posteriormente probar y verificar el correcto funcionamiento de los controles diseñados.

  En el caso del modelado de la relación entre la fuerza vertical ejercida por las hélices y la entrada PWM se dispone de la ecuación 4.14. Como se desconoce la constante C_t que es dependiente de la geometría de la hélice, se sustituyen los términos $\frac{1}{4C_t\rho\pi R^4}$ por la constante K_{td} que será ajustada mediante pruebas controladas de vuelo. Estas pruebas consisten en realizar una fuerza con las hélices igual al peso del quadrotor, modificando K_{td} hasta conseguir que el dron quede aproximadamente estático en el aire. La constante finalmente obtenida resulta $K_{td} = 29000$.

  Para el ajuste de las velocidades angulares del autopiloto, se realiza un montaje que permite la rotación del ángulo deseado a modelar, bloqueando las de los otros dos ángulos (Figura 6.1). Mediante el envío de distintas señales PWM se mide con el IMU la velocidad angular resultante, utilizando todas estas medidas para realizar el ajuste de una ecuación que relaciona directamente la velocidad angular con el PWM a aplicar de la forma:

  $$PWM = K_{exp}\omega_d + PWM_0$$

  donde K_{exp} es la constante proporcional ajustad, ω_d es la velocidad deseada y PWM_0 es el punto en el que se centra el PWM del canal correspondiente.

El resultado obtenido finalmente en estas pruebas ha sido satisfactorio, consiguiendo un modelo bastante aproximado del comportamiento del autopiloto. Las ecuaciones ajustadas se detallan en el apéndice A.

- **Ajuste del control angular**: Una vez modelado el autopiloto y tras numerosas simulaciones y verificaciones del correcto funcionamiento de los controles diseñados, del generador de trayectorias y del navegador implementados, se procede a la verificación de su comportamiento en el dron real de su comportamiento. Pero para poder verificar todo correctamente, primero es necesario ajustar bien los controles diseñados. Tras muchas pruebas y comprobaciones de la respuesta de éstos, se han seleccionado los explicados en la sección 4.5.

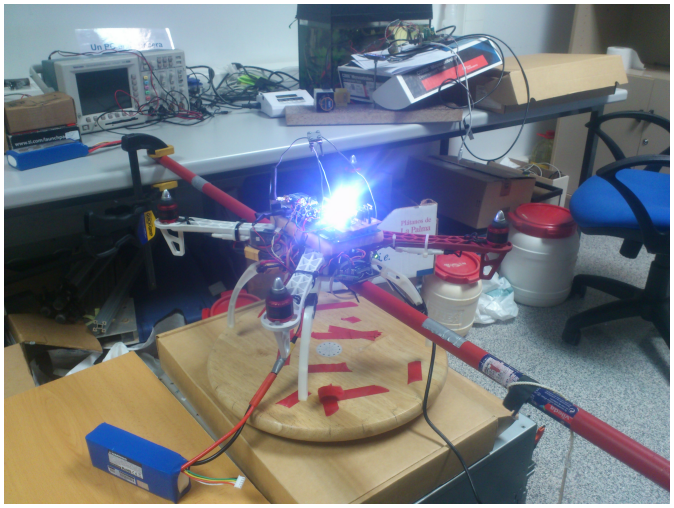

Figura 6.1: Montaje realizado para modelar y probar el control de los ángulos.

Se ha procedido del mismo modo que para el modelado del autopiloto, permitiendo sólo el giro que se desea controlar, bloqueando los otros dos y el movimiento en el eje Z en el caso del control angular. En la figura 6.2 se puede observar la respuesta del ángulo pitch ante una consigna de 5 grados. El tiempo de respuesta en el que el sistema alcanza la referencia es muy pequeño, inferior a 1 segundo, permitiendo corregir en muy poco tiempo las posibles desviaciones provocadas por el aire o movimientos no deseados. Las variaciones alrededor de la consigna se deben a la precisión del IMU, el cual oscila alrededor de medio grado sobre el ángulo real provocando un desvío en el control, del mismo modo que al no incluir un integrador, el sistema no es capaz de corregir las desviaciones derivadas de la aproximación del modelo realizado.

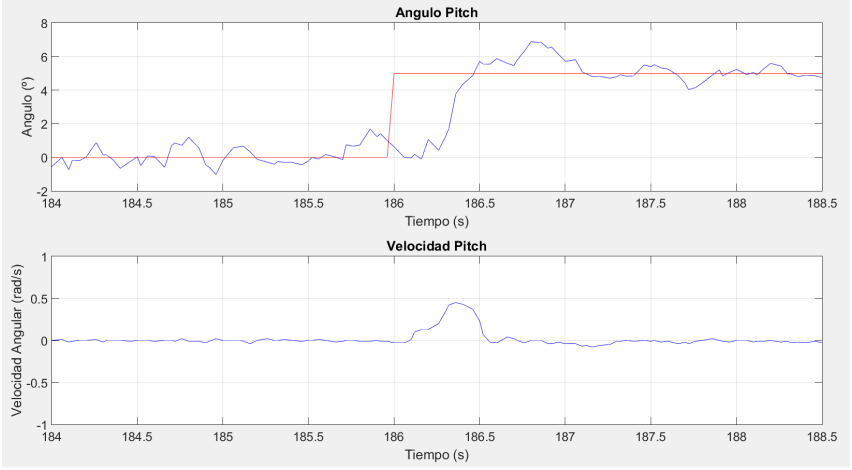

Figura 6.2: Respuesta del ángulo pitch (azul) ante un escalon de 5 grados (rojo).

En la siguiente figura se observa la respuesta del ángulo pitch ante tres referencias de -10, 10 y 0 grados con el generador de trayectorias y el navegador integrados:

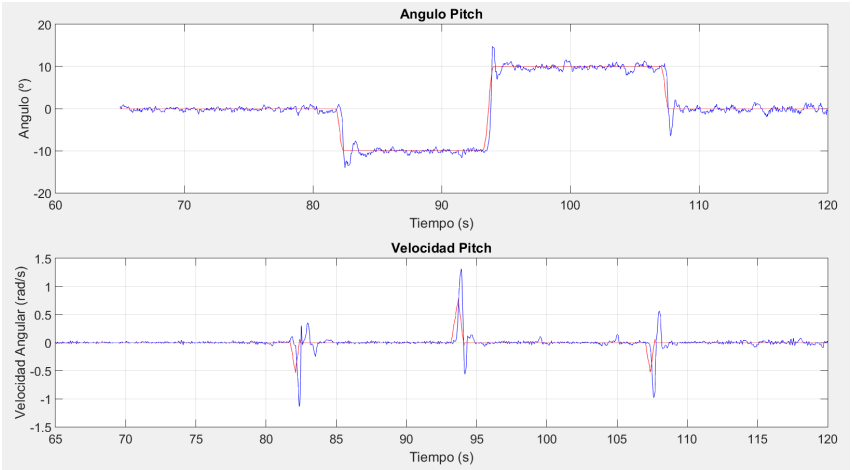

Figura 6.3: Respuesta del ángulo pitch (azul) ante una serie de referencias (rojo).

Los sobrepasamientos del ángulo al alcanzar la referencia son debidos al control de velocidad que integra el autopiloto, el cual necesita un ajuste de ganancias para mejorar la respuesta. Con la inclusión del generador de trayectorias, se consigue reducir este efecto al limitar las acciones realizadas, que en el caso de haber una variación muy grande de la consigna influye en mayor medida. Respecto de la oscilación alrededor de la medida, el sistema se ve afectado de la misma forma que en el caso anterior, siendo el error asumible dada la precisión del control.

- **Ajuste del control de altura**:

  En el caso del control de altura, se ha limitado la altura máxima a la que puede ascender el dron mediante el uso de cuerdas atadas en las patas. De este modo, en caso de fallo

las posibilidades de que haya un accidente disminuyen mucho, puesto que el propio vuelo del quadrotor es muy peligroso debido a la potencia de los motores y las hélices. En la siguiente imagen se muestra una prueba realizada para verificar el correcto funcionamiento del control, sujetando con una pértiga el dron (sin levantarlo con ella) para evitar que en caso de fallo se descontrole:

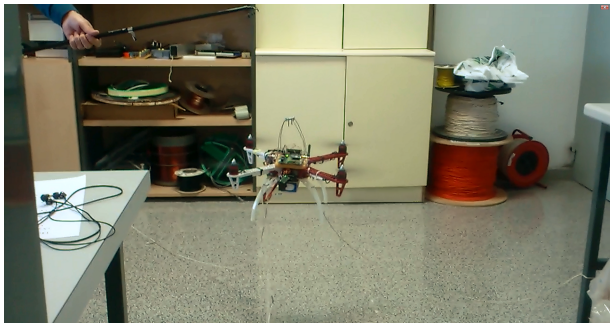

Figura 6.4: Montaje realizado para probar el funcionamiento del control de altura.

En la siguiente figura se muestra la respuesta del control de altura con un tiempo de respuesta calculado de 3 segundos con un periodo de control de 100 ms. Se puede observar que la altura se mantiene casi constante, oscilando debido a la precisión del sensor de altura. Ésta es de 1 cm, por lo que la velocidad mínima que se puede detectar a partir de este sensor será de 10 cm/s.

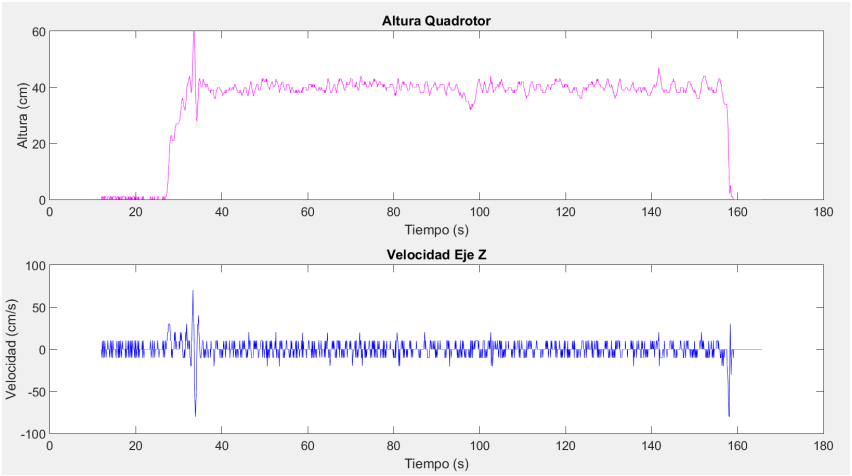

Figura 6.5: Respuesta del control de altura para una consigna de 40 cm.

Capítulo 7

Conclusiones

Con la realización de este Trabajo de Final de Grado, se ha buscado tener una primera aproximación a la navegación y control de misión de los drones, concretamente de quadrotores. Por tanto, las investigaciones efectuadas, ya sean con un resultado satisfactorio o no, sirven como base para su futuro desarrollo.

Se ha diseñado un sistema de planificación, generación de trayectorias y control de misión complejo, que es capaz de ejecutar varios tipos de movimientos y seguir una ruta planificada. Esto ha sido simulado exhaustivamente en Matlab para asegurar su funcionamiento y posteriormente se ha implementado en el dsp seleccionado. También se ha desarrollado una interfaz gráfica que permite la interacción con el dron de una forma simple, permitiendo la visualización de la telemetría del dron en tiempo real y la modificación de las consignas a alcanzar.

Para la verificación de los programas diseñados, se ha desarrollado un entorno de simulación en 3 dimensiones que permite ver el comportamiento real del quadrotor, pudiendo definir un espacio cerrado para la navegación para simular las medidas que devolverían los sensores de posición. Esto facilita las tareas de simulación y puesta a punto de las técnicas de navegación, y podrá ser utilizado en un futuro para testar de forma simple la navegación en entornos cerrados y el evitar obstáculos. Esto último fue inicialmente planteado como un objetivo a conseguir. Se realizó un estudio básico sobre ello, pero fue finalmente descartado el diseño e implementación debido a la magnitud del trabajo total.

A pesar de diseñar un sistema de navegación que permita moverse por el plano horizontal XY, no ha sido posible la implementación final sobre el quadrotor real para realizar las trayectorias complejas de los controles necesarios debido a la baja precisión y error acumulativo del sensor inercial IMU. Se han dedicado numerosas horas de trabajo a la calibración, filtrado y mejora de las medidas que éste proporciona, pero no se ha conseguido obtener una medida fiable de la aceleración, necesaria para el cálculo de la velocidad de avance y posición del quadrotor. Por tanto, se plantea como una mejora futura el desarrollo de un sistema de medición inercial con el se que pueda navegar por interiores con las técnicas desarrolladas en este proyecto y validadas en simulación. También se podrá utilizar en exteriores un GPS, que permitiría realizar las trayectorias complejas de seguimiento sin necesidad de hacer una mejora en el IMU.

Por el contrario, los controles de orientación y altura han sido implementados de forma exitosa pudiendo efectuar maniobras de despegue y aterrizaje de forma estable, e incluso avances frontales y laterales variando los ángulos roll y pitch.

Debido a la complejidad del software a diseñar, a los problemas derivados de la integración del hardware y a la propia dinámica del quadrotor, este TFG ha requerido gran cantidad de tiempo y esfuerzo para conseguir la implementación final del sistema desarrollado y realizar pruebas de vuelo para verificar el correcto funcionamiento del trabajo hecho. Para ésto, ha sido necesario programar alrededor de 20000 líneas de código para efectuar las pruebas de cada sensor y test específicos, de las que alrededor de 10000 forman parte del programa final.

Como conclusión personal diré que el desarrollo de este TFG me ha permitido adquirir una mayor experiencia con los sistemas reales, complementando a su vez todo lo aprendido en la carrera.

Apéndice A

Magnitudes físicas del quadrotor

En este apéndice se recogen las magnitudes físicas que intervienen en el modelo simplificado del quadrotor y las funciones que se han ajustado para controlar el autopiloto mediante las entradas PWM.

El valor de las constantes de los modelos utilizados se detalla en la siguiente tabla:

Magnitud	Valor	Unidad
b_x	0.1	kg/s
b_y	0.1	kg/s
b_z	0.1	kg/s
m	1.55	kg
g	9.82	m/s^2
ρ	1,185	Kg/m^3
R	0.13	m

Cuadro A.1: Magnitudes físicas utilizadas.

Por otra parte, se ha realizado el ajuste de tres funciones que permitan relacionar la velocidad angular que se desea aplicar con el PWM que produce esa velocidad. Las ecuaciones obtenidas son las siguientes:

$$PWM_{Roll} = K_R\omega_{Roll} + PWM_{R0}$$

$$PWM_{Pitch} = K_P\omega_{Pitch} + PWM_{P0}$$

$$PWM_{Yaw} = K_Y\omega_{Yaw} + PWM_{Y0}$$

siendo K_R, K_P y K_Y las constantes proporcionales ajustadas y PWM_{R0}, PWM_{R0} y PWM_{R0} los puntos centrales sobre los que se centra cada canal PWM.

El valor de las constantes ajustadas es:

Constante	Valor
K_R	127.1345
K_P	-127.1345
K_Y	-210.6049
PWM_{R0}	1782.5
PWM_{P0}	1782.5
PWM_{Y0}	1779.6

Cuadro A.2: Constantes ajustadas para el control angular.

Para hallar el valor de la constante que relaciona la velocidad angular de las hélices y el PWM generado, se utiliza la relación de velocidad de los motores síncronos utilizados en el quadrotor, que está determinada por la siguiente ecuación:

$$\omega_{motor}(rpm) = K_v V_{in} \tag{A.1}$$

La variable K_v es la constante que relaciona la velocidad angular del motor (en rpm) con la tensión aplicada a éste. La proporciona el fabricante del motor, y en esta caso tiene un valor de $K_v = 920\ rpm/V$. Reescribiendo la ecuación, se obtiene la tensión que se necesita para hacer una cierta velocidad angular ω_{motor}:

$$V_{in} = \frac{\omega}{K_v} \tag{A.2}$$

De este modo, la velocidad máxima que pueden hacer los motores está limitada por la tensión de la batería V_{bat}, que en este caso es una batería de iones de litio de 4 celdas que proporciona un máximo de 16.8 V. Establecemos por tanto el porcentaje del PWM aplicado en función de la tensión de la batería:

$$PWM(\ \%) = \frac{\omega}{920}\frac{1}{V_{bat}} \tag{A.3}$$

Como se desea obtener un valor entre 1 y 2 ms, basta con multiplicar $x10$ la ecuación anterior para obtener el tiempo en activo de la señal PWM entre 1000 y 2000 μs:

$$t(\mu s) = 1000\frac{\omega}{920}\frac{1}{V_{bat}} \tag{A.4}$$

Al estar invertido el canal en la emisora, 2 ms equivalen a la potencia mínima y 1 ms a la máxima, por lo que se resta a 2 ms la ecuación anterior para invertirlo:

$$t(\mu s) = 2000 - 1000\frac{\omega}{920}\frac{1}{V_{bat}} \tag{A.5}$$

Realizando un escalado para las equivalencias temporales con los valores del *Output Compare*,

se obtiene finalmente la ecuación:

$$PWM_{Throttle} = 2344 - \omega \frac{30}{\pi} \frac{1172}{920 V_{bat}} \tag{A.6}$$

Tomando los valores $\frac{30}{\pi}\frac{1172}{920}$ como la constante K_{PwmT} y sustituyendo la velocidad angular por la de la ecuación 4.14, obtenemos la expresión final:

$$PWM_{Throttle} = 2344 - K_{KpwmT}\, TK_{td} \frac{1}{920 V_{bat}} \tag{A.7}$$

siendo los valores $K_{KpwmT} = 12{,}144$ y K_{td} el ajustado en la sección 6.

Bibliografía

[1] 3DR. 3dr power module. URL http://copter.ardupilot.com/wiki/common-3dr-power-module/.

[2] digi. Xbee-pro 868 rf modules. URL http://www.digi.com/products/xbee-rf-solutions/modules/xbee-pro-868.

[3] DJI. F450 user manual. . URL http://www.dji.com/product/flame-wheel-arf/feature.

[4] DJI. Naza m-lite user manual. . URL http://www.dji.com/product/naza-m-lite/feature.

[5] Beatriz Frisón. Modelado y control de un helicóptero de cuatro motores. *Proyecto Final de Carrera. Universidad de Zaragoza, CPS*, 2008/2009.

[6] Futaba. Futaba 6j manual. URL http://www.futabarc.com/systems/futk6000.html.

[7] Texas Instruments. Sys/bios. . URL http://processors.wiki.ti.com/index.php/Category:SYSBIOS.

[8] Texas Instruments. Tms320f2812 data manual. . URL http://www.ti.com/product/TMS320F2812.

[9] Katsuhiko Ogata. *Ingeniería de Control Moderna, 5ª edición*. PEARSON EDUCACIÓN, S.A., 2010.

[10] Parallax. Ping documentation. URL https://www.parallax.com/product/28015.

[11] Pololu. Pololu rc switch user's guide. URL https://www.pololu.com/product/2806.

[12] Sparkfun. Ahrs code. URL https://www.sparkfun.com/products/10736.

[13] John A. Stankovic. Misconceptions about real-time computing. *IEEE Computer Society*, 1988.

[14] The UAV. 2014. URL http://www.theuav.com/.