



Universidad
Zaragoza

Trabajo Fin de Grado

DESARROLLO E IMPLEMENTACIÓN DE UN CUADRICÓPTERO

Autor

Eugenio Alcalá Baselga

Director

Alfonso Blesa Gascón

Escuela universitaria politécnica de Teruel
2014

DESARROLLO E IMPLEMENTACIÓN DE UN CUADRICOPTERO

RESUMEN

El presente trabajo constituye un proyecto de desarrollo de un prototipo de cuadricóptero haciendo uso de los conocimientos adquiridos durante los cuatro años de grado.

Se ha logrado elaborar un prototipo ligero con un sistema de control de estabilidad automático que se comunica vía Wi-Fi con el punto de control remoto pudiendo ser este un Smartphone o un PC.

Se ha utilizado como plataforma de control la placa Raspberry Pi. La aplicación se ha desarrollado sobre la plataforma COSME la cual trabaja sobre el sistema operativo modificado con características de tiempo real.

La estructura se ha fabricado con materiales resistentes obteniendo así una rigidez importante para un primer prototipo como es este.

Utilizando sensores inerciales se ha implementado un algoritmo para el cálculo de los ángulos de vuelo corrigiendo las desventajas que aportan a su vez los sensores y se han calibrado los motores obteniendo así un factor constante diferente en cada motor que nos permite igualar en potencia los cuatro motores para un óptimo funcionamiento.

La regulación ha sido abordada haciendo uso de métodos para conseguir una buena fiabilidad en vuelo.

INDICE GENERAL

1. Introducción.....	1
a. Objetivos.....	2
i. Objetivo general	2
ii. Objetivos específicos	2
b. Cronograma del proyecto.....	2
2. Desarrollo teórico	4
a. Vehículos aéreos no tripulados (UAV o VANT).....	4
b. Cuadricóptero.....	5
c. Dinámica del cuadricóptero.....	6
i. Rotación sobre los ejes X e Y	7
ii. Rotación sobre el eje Z.....	8
iii. Control de altitud	10
d. Estructura	11
i. Peso del cuadricóptero.....	11
e. Raspberry Pi.....	12
i. Introducción.....	12
ii. Hardware	12
iii. Software.....	14
iv. Entorno de programación y configuración	16
f. IMU.....	16
i. Acelerómetro (ADXL345).....	18
ii. Giróscopo (ITG3200).....	19
iii. Magnetómetro (HMC5883L).....	20
g. Microcontrolador PCA9685	22

h. Actuadores.....	24
i. Motores brushless	24
ii. Variadores de velocidad (ESC)	26
i. Alimentación.....	28
j. Comunicación	29
k. Procesado de las señales de la IMU.....	30
ii. Manipulación de los datos.....	30
k. Control de estabilidad.....	37
i. Controlador PID	37
3. Desarrollo e implementación del COSMEcopter.....	40
a. Esquema hardware	40
b. Aplicación en COSME.....	41
c. Desarrollo de los bancos de pruebas.....	45
d. Diseño de circuitos impresos.....	48
i. Placa de alimentación.....	48
ii. Circuito impreso de sensado	49
e. Pruebas	52
i. Puesta en marcha y calibración de los sensores	52
ii. Calibración de los 4 motores. Curvas de potencia	56
iii. Calibración del control PID	58
iv. Consumos del cuadricóptero	60
4. Componentes y coste	62
5. Conclusiones y futuro	63
a. Conclusiones	63
b. Trabajo futuro.....	64

6. Agradecimientos.....	65
BIBLIOGRAFÍA	66
ANEXOS.....	68
Anexo A: Galería fotográfica.....	68
Anexo B: Datasheet de los sensores y el microcontrolador.....	72
Anexo C: Script de Matlab para la calibración de los motores	72
Anexo D: Código de la aplicación	72

INDICE DE FIGURAS

Figura 1: Diagrama de gantt del proyecto.....	3
Figura 2: Geometría de la estructura	5
Figura 3: Sentido de giro de los motores del cuadricóptero.....	6
Figura 4: Comportamiento cuando varía el ángulo roll	7
Figura 5: Comportamiento cuando varía el ángulo pitch.....	8
Figura 6: Comportamiento cuando varía el ángulo yaw	10
Figura 7: Comportamiento de los motores para subir a lo largo del eje Z.....	10
Figura 8: Orden de ejecución de las instancias	15
Figura 9: Representación del bloque funcional.....	15
Figura 10: Reverso de la IMU	17
Figura 11: Definición del sistema de referencia	17
Figura 12: IMU	17
Figura 13: Diagrama de bloques del acelerómetro	18
Figura 14: Vista en planta con cotas del acelerómetro.....	19
Figura 15: Diagrama de bloques del giróscopo	20
Figura 16: Representación de los ejes del giróscopo	20
Figura 17: Diagrama de bloques y conexiones del magnetómetro	21
Figura 18: Vista en planta con cotas del magnetómetro	21
Figura 19: Representación de la señal PWM.....	22

Figura 20: Diagrama de bloques del PCA9685	23
Figura 21: Situación del microcontrolador PCA9685 en la expansion board.....	23
Figura 22: Motor brushless DJI	25
Figura 23: Alzado y planta del motor con cotas	25
Figura 24: Funcionamiento de la hélice.....	26
Figura 25: Variador DJI.....	27
Figura 26: Dongle Wi-Fi	29
Figura 27: Proyecciones de los vectores de un sistema de referencia (refs) sobre el otro sistema (refc)	31
Figura 28: Comprobación de la simplificación	32
Figura 29: Obtención de los ángulos de Euler	35
Figura 30: Diagrama de bloques funcional del método	36
Figura 31: Esquema del regulador PID en paralelo	39
Figura 32: Esquema eléctrico del hardware	40
Figura 33: Diagrama de bloques funcionales de la aplicación	42
Figura 34: Instrumento para la medición del empuje.....	46
Figura 35: Banco de calibración.....	47
Figura 36: Sistema de agarre con rodamiento	47
Figura 37: Fichero board de la placa distribuidora de energía.....	48
Figura 38: Circuito de distribución de energía	49
Figura 39: Fichero schematic del circuito de sensado.....	50

Figura 40: Fichero board del circuito de sensado	51
Figura 41: Circuito de sensado sin componentes.....	51
Figura 42: Circuito de sensado con componentes soldados	52
Figura 43: Representación ángulos roll y pitch en Matlab.....	54
Figura 44: Ángulo pitch en MiniBlas.....	55
Figura 45: Ángulo roll en MiniBlas.....	55
Figura 46: Gráfica empuje frente a PWM.....	56
Figura 47: Grafica empuje frente a PWM aplicado un factor de corrección a la curva de cada motor	57
Figura 48: Sistema críticamente estable en miniBlas.....	59

INDICE DE TABLAS

Tabla 1: Pesos por componente y total del cuadricóptero	11
Tabla 2: Tabla de Ziegler-Nichols.....	59
Tabla 3: Tabla de consumos y potencias	61
Tabla 4: Cálculo de la autonomía	61
Tabla 5: Lista de componentes con sus respectivos costes	62

INDICE DE ECUACIONES

Ecuación 1: Fuerza de arrastre	9
Ecuación 2: Fuerza de sustentación	9
Ecuación 3: Matriz de rotación.....	32
Ecuación 4: Matriz de rotación instantánea.....	32
Ecuación 5: Matriz de rotación en el instante (t + dt).....	33
Ecuación 6: Matriz de rotación en el instante inicial	33
Ecuación 7: X e Y ortogonales.....	34
Ecuación 8: Z ortogonal	34
Ecuación 9: Vectores normalizados.....	35
Ecuación 10: Expresión general del regulador PID.....	38
Ecuación 11: Expresión del PID no lineal.....	44
Ecuación 12: Calculo del pwm en el control de estabilidad	45
Ecuación 13: Expresiones para el cálculo de los ángulos de vuelo	53
Ecuación 14: Relación entre tiempos integral y derivativo y la constante proporcional	60

NOTACIÓN

ADC: Conversor Analógico Digital (del inglés *Analog-to-Digital Conversion*)

COSME: Control System and Modelling Environment

DCM: Matriz de Cosenos Directores (del inglés *Matrix of Direction Cosines*)

ESC: Controlador Electrónico de Velocidad (del inglés *Electronic Speed Controller*)

GND: Tierra (del inglés *ground or earth*)

GPIO: Entradas-Salidas de propósito general (del inglés *General Purpose Input/Output*)

I²C: Inter-Integrated Circuit

IMU: Unidad de Medición Inercial (del inglés *Inertial Measurement Unit*)

Li-Po: Polímero de Litio

PWM: Modulación por ancho de pulso (del inglés *Pulse Width Modulation*).

PID: Proporcional Integral Derivativo (del inglés *proportional-integral-derivative*).

SPI: Serial Peripheral Interface

SDA: Datos (del inglés *Source Data*)

SCL: Reloj (del inglés *Source Clock*)

SSH: Secure Shell

UART: Transmisor-Receptor Asíncrono Universal (en inglés *Universal Asynchronous Receiver-Transmitter*)

UAV: Vehículo Aéreo No Tripulado (en inglés *Unmanned Air Vehicle*).

VANT: Vehículo Aéreo No Tripulado

Wi-Fi: Wireless Fidelity

FB: Bloque Funcional (del inglés *Functional Block*)

1. Introducción

Actualmente en el campo de la robótica existen muchas áreas de la ingeniería pero la aeronáutica con especial presencia, sobre todo con el surgimiento de los UAVs desarrollados para un gran abanico de tareas, desde el entorno militar, hasta equipos de investigación.

En el presente proyecto se pretende elaborar un cuadricóptero, de peso ligero y de fácil montaje, el cual consiste en una estructura rígida en forma de cruz, con un motor en cada extremo de la estructura. El cuadricóptero será controlado a través de la plataforma Raspberry Pi dotada de características de tiempo real.

Dentro de las principales razones por las cuales se toma la decisión de abordar un proyecto como es este, es por la gran cantidad de áreas involucradas en el desarrollo de un prototipo de estas características, permitiendo la oportunidad de aplicar y demostrar una gran cantidad de conocimientos que se han adquirido en el grado. Entre las áreas de estudio que se pretenden aplicar se pueden mencionar: programación, administración avanzada de Linux, microcontroladores, programación en lenguaje C, sistemas de control automático, sistemas de comunicación, máquinas eléctricas, diseño de circuitos, mecánica y física.

a. Objetivos

i. Objetivo general

Diseñar y construir un vehículo aéreo no tripulado con un control de estabilidad automático utilizando como plataforma de control la Raspberry Pi y COSME.

ii. Objetivos específicos

1. Esbozar y construir una estructura mecánica ligera acorde a la potencia de los motores.
2. Desarrollar un sistema de sensado haciendo uso de un giróscopo, acelerómetro y magnetómetro.
3. Elaborar el software del UAV sobre la plataforma COSME trabajando en tiempo real.
4. Diseñar e implementar un control que se encargue de la estabilidad del sistema.
5. Integrar todas las tecnologías implicadas construyendo un prototipo operativo.

b. Cronograma del proyecto

El cronograma creado inicialmente ha sufrido modificaciones a lo largo del proyecto quedando finalmente de la siguiente manera:

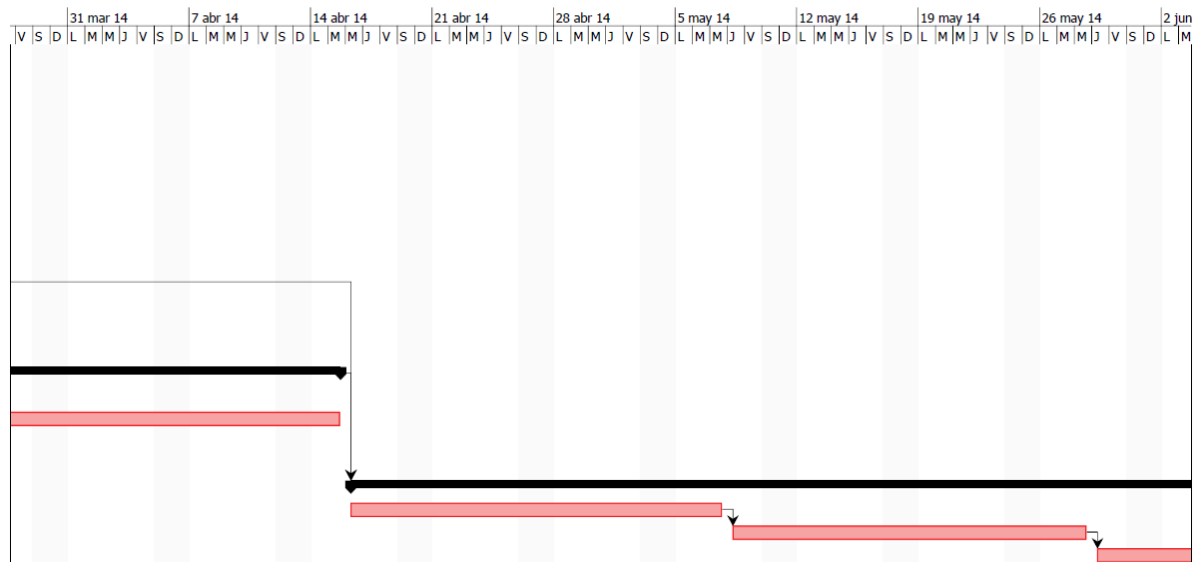
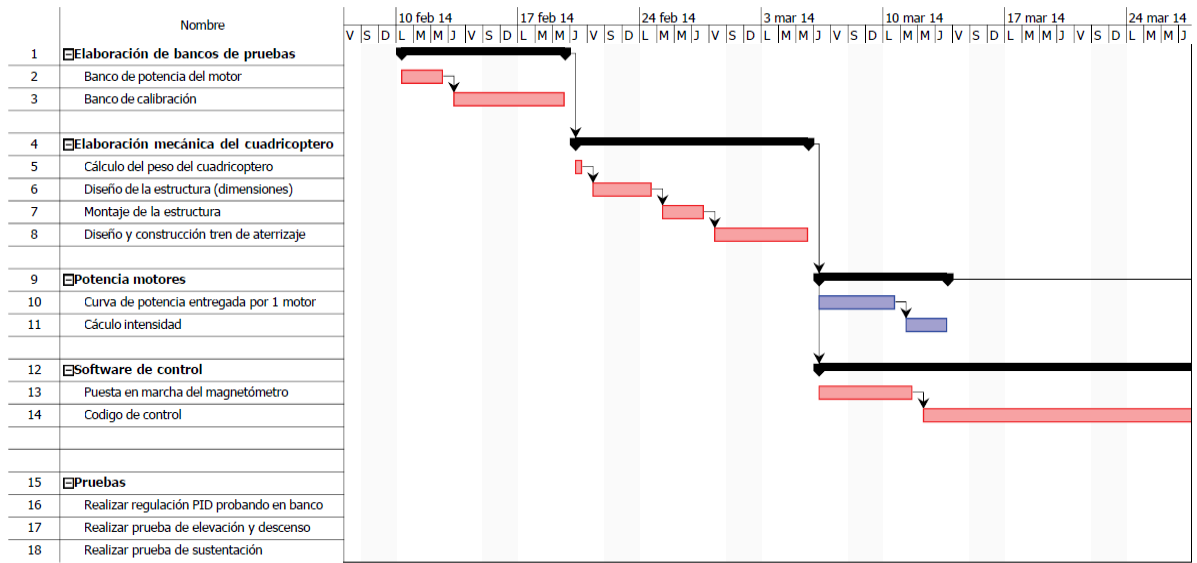


Figura 1: Diagrama de gantt del proyecto

2. Desarrollo teórico

En el desarrollo teórico, se sintetizarán los temas más relevantes para el desarrollo y entendimiento de los conceptos necesarios para la implementación del prototipo del cuadricóptero.

a. Vehículos aéreos no tripulados (UAV o VANT)

Un vehículo aéreo no tripulado, en inglés Unmanned Aerial Vehicle, o también conocido como drone, es una aeronave que vuela sin tripulación.

Existen algunos UAV de uso civil aunque la gran mayoría son usados en aplicaciones militares, donde son denominados vehículo aéreo de combate no tripulado.

Históricamente los UAV eran simplemente aviones pilotados remotamente pero cada vez se está empleando más el control autónomo de estos. Existen dos variantes: algunos son controlados desde una posición remota y otros funcionan de forma autónoma.

Actualmente, los UAV militares realizan tanto misiones de reconocimiento como de ataque. También son utilizados en aplicaciones civiles, como en labores de lucha contra incendios o seguridad civil, como la vigilancia de los oleoductos. Los vehículos aéreos no tripulados suelen ser preferidos para misiones que son demasiado "aburridas, sucias o peligrosas" para los aviones tripulados.

Los UAV se distinguen por infinidad de factores entre otros la autonomía, el tamaño, el peso, la finalidad y el tipo de ala. En función del tipo de ala podemos encontrar los aviones (ala fija) y los helicópteros y multicópteros. Estos últimos se pueden dividir en bicópteros, tricópteros, cuadricópteros, hexacópteros, octacópteros y otros multicópteros.

b. Cuadricóptero

A diferencia de la mayoría de los helicópteros, los cuadricópteros incorporan una hélice en cada uno de los cuatro motores, los cuales normalmente están situados simétricamente en los extremos de los brazos de la estructura (figura 2).

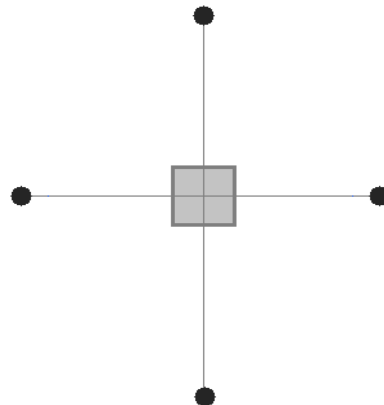


Figura 2: Geometría de la estructura

Utilizan un sistema de control electrónico y sensores electrónicos, situados sobre el eje del centro de gravedad del UAV, para estabilizar la aeronave. El control de movimiento del vehículo se consigue mediante la alteración de la velocidad de rotación (rpm) de uno o más motores, cambiando de ese modo su carga de par y las características de

empuje / elevación. Con su pequeño tamaño y la maniobrabilidad ágil pueden ser volados tanto en interiores como en exteriores.

c. Dinámica del cuadricóptero

Uno de los conceptos que surge cuando se empieza a estudiar el campo de la aerodinámica es el de la fuerza de sustentación. La sustentación es la fuerza generada sobre un cuerpo que se desplaza a través de un fluido (el aire), de dirección perpendicular a la de la velocidad de la corriente incidente. El paso del fluido por la superficie del objeto crea una fuerza con varios componentes, la fuerza de sustentación y entre otras la fuerza de arrastre, la cual es la encargada de crear un par rotacional cuya dirección va en contra del sentido de giro de las hélices. Por esta razón es que los helicópteros poseen rotores de cola que compensan este par rotacional y evita que giren sin control alrededor del eje de rotación del rotor principal.

En el caso del cuadricóptero dos motores giran en el sentido de las agujas del reloj y los otros dos en sentido contrario a las agujas del reloj pudiendo así contrarrestar el par rotacional provocado por cada motor.

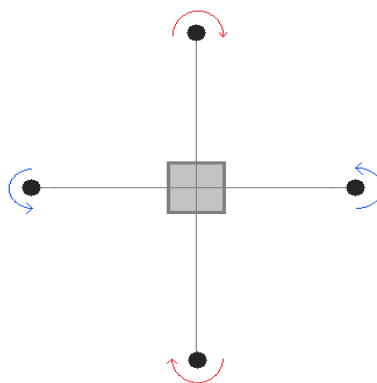


Figura 3: Sentido de giro de los motores del cuadricóptero

La dinámica de los cuadricópteros cuenta con 3 tipos distintos de movimientos básicos, los cuales se pueden ejecutar en conjunto para lograr el tipo de movimiento deseado. Estos tipos de movimientos son las rotaciones sobre los propios ejes del cuadricóptero y se explican a continuación.

i. Rotación sobre los ejes X e Y

La dinámica de rotación es igual para ambos ejes. A la rotación sobre el eje X se le denomina roll y a la rotación sobre el eje Y se le denomina pitch. Para realizar una rotación sobre uno de los ejes es necesario aumentar la velocidad de giro de un motor y disminuir la velocidad de giro del opuesto.

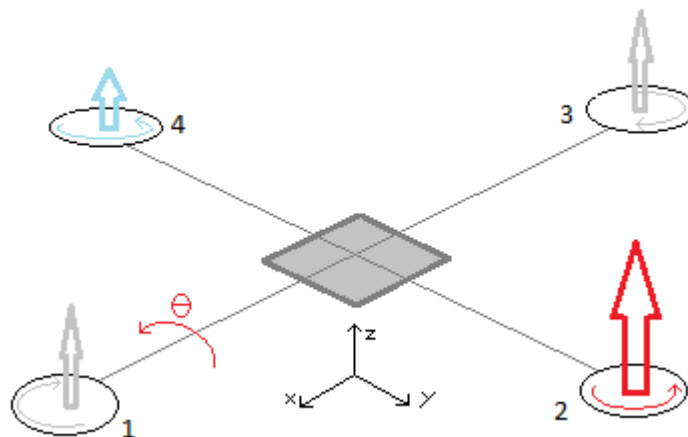


Figura 4: Comportamiento cuando varía el ángulo roll

De este modo, si la variable que se quiere modificar es el ángulo roll (Θ) las velocidades angulares de los motores 1 y 3 permanecerán constantes y las velocidades angulares de los motores 2 y 4 cambiarán en función de si el ángulo roll es positivo o negativo (figura 4).

Por otro lado se ha de mantener en todo momento la misma variación de potencia para los dos motores, es decir, si a un motor se le aplica una variación $+\Delta\omega$ al opuesto se le deberá aplicar $-\Delta\omega$ para evitar aumentar el par rotacional de la estructura.

La dinámica resulta análoga para el caso de la rotación en torno al eje Y (figura 5).

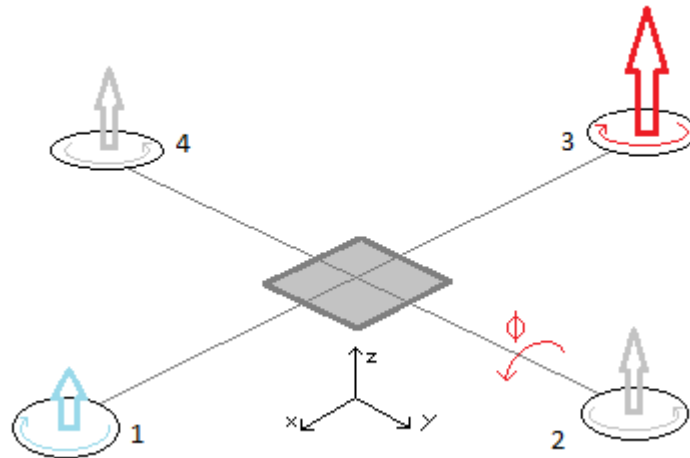


Figura 5: Comportamiento cuando varía el ángulo pitch

Como consecuencia de este ángulo de pitch o de roll el cuadricóptero experimentará un desplazamiento. Por ejemplo, si se produce una rotación positiva sobre el eje Y el cuadricóptero se desplazará en la dirección X positiva.

ii. Rotación sobre el eje Z

El eje Z atraviesa de forma vertical el cuadricóptero pasando por el centro de gravedad de este. Manipulando la velocidad de rotación de los distintos motores se puede llegar a un punto de operación donde el par rotacional neto del UAV sea igual a cero.

Tanto la fuerza de sustentación como la fuerza de arrastre son funciones que dependen del cuadrado de la velocidad angular. A continuación se muestran las ecuaciones de dichas fuerzas:

$$F_{arrastre} = \frac{1}{2} \rho C_d \omega_i^2$$

Ecuación 1: Fuerza de arrastre

$$F_{sustentación} = \frac{1}{2} \rho C_T \omega_i^2$$

Ecuación 2: Fuerza de sustentación

, donde C_d y C_T son los coeficientes de arrastre y sustentación respectivamente, ρ es la densidad del aire y ω_i es la velocidad angular de cada motor.

Por lo tanto, si se mantienen constantes el tipo de fluido donde se está volando, y el tipo de hélices que se están utilizando, la única variable que queda es la velocidad angular.

Llegados a este punto, como de lo que se trata es de girar en torno al eje Z lo único que se debe hacer es aplicar una velocidad angular mayor a la pareja de motores que giran en un sentido y disminuir la velocidad angular de los otros dos. Si aumenta la velocidad angular de los motores que giran en el sentido de las agujas del reloj el cuadricóptero girará hacia la derecha, en el caso contrario el cuadricóptero girará hacia la izquierda.

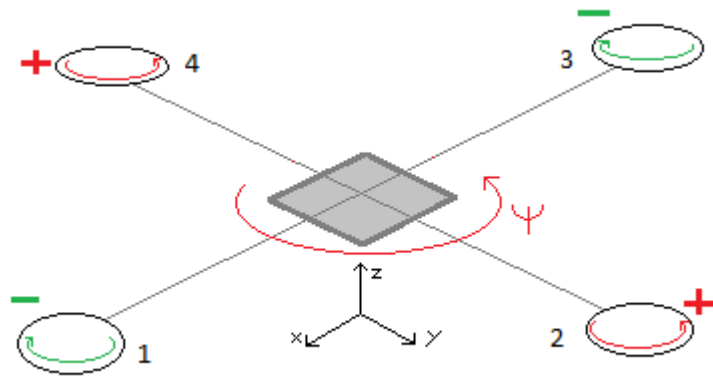


Figura 6: Comportamiento cuando varía el ángulo yaw

iii. Control de altitud

El control de altitud se realiza aumentando o disminuyendo la velocidad angular de todos los motores de manera equivalente.

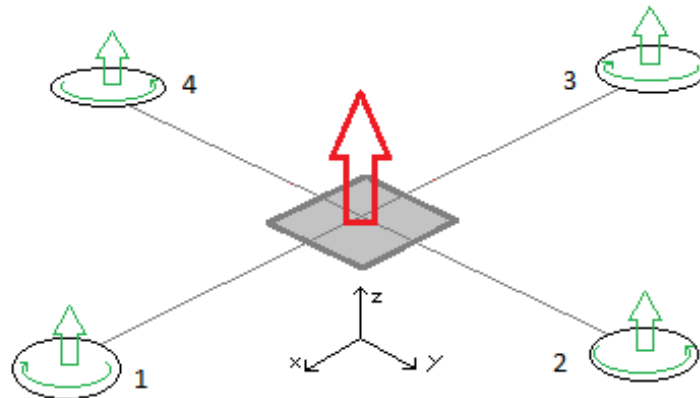


Figura 7: Comportamiento de los motores para subir a lo largo del eje Z

d. Estructura

El esqueleto del UAV está formado por perfiles de aluminio de 15 x 15 x 1 mm formando entre ellos un ángulo de 90°. Cada brazo mide 25 cm y sobre ellos, en la intersección, se encuentra la base sobre la que van montados los componentes electrónicos, esta base es de metraquilato de 4 mm de espesor.

i. Peso del cuadricóptero

La siguiente tabla muestra los componentes que forman el cuadricóptero y sus respectivos pesos.

Material	Peso (g)
Aluminio	140
Poliestireno	142
Bateria	278
Placa distribución energía	20
Placa sujección batería	6
Raspberry Pi + PCA9685 + dongle	104
3 sensores	2
ESC's	84
Motores	212
Hélices	24
Otros	55
PESO TOTAL EN GRAMOS	1067

Tabla 1: Pesos por componente y total del cuadricóptero

e. Raspberry Pi

i. Introducción

Raspberry Pi es un ordenador de bajo costo, desarrollado en Reino Unido por la Fundación Raspberry Pi. El diseño no incluye un disco duro, usa una tarjeta SD para el almacenamiento permanente.

La fundación da soporte para las descargas de las distribuciones para arquitectura ARM, Raspbian, RISC OS 5, Arch Linux ARM y Pidora; y promueve principalmente el aprendizaje del lenguaje de programación Python, y otros lenguajes como Tiny BASIC, C y Perl.¹

ii. Hardware

El diseño incluye un System-on-a-chip (SOC) Broadcom BCM2835, que contiene un procesador central (CPU) ARM1176JZF-S a 700 MHz, un procesador gráfico (GPU) VideoCore IV, y 512 MiB de memoria RAM compartidos con la GPU.

Incorpora 2 puertos USB 2.0 de los cuales uno de ellos será utilizado para la comunicación vía Wi-Fi, entrada de video que permite instalar un módulo de cámara, salida de video tanto RCA como HDMI y salida de audio.

El diseño no incluye un disco duro, usa una tarjeta SD para el almacenamiento permanente e incorpora un controlador Ethernet para la conexión a red.

¹ Información extraída de http://es.wikipedia.org/wiki/Raspberry_Pi

El puerto GPIO (General Purpose Input Output) consta de 26 pines entre los cuales tenemos los referentes a la comunicación I²C, UART y SPI, el pin de 3.3 V, el de 5V y el pin GND.²

El consumo energético es de 3.5 W ya que necesita 700mA a 5 V, a esto hay que sumarle el consumo de periféricos como el dongle Wi-Fi.

Cabe resaltar la comunicación I²C ya que ha sido la utilizada en el proyecto para la relación de los sensores con la unidad de control (Raspberry Pi). Su nombre viene de Inter-Integrated Circuit (Circuitos Inter-Integrados). La velocidad de transmisión es de 100 kbits por segundo y su principal característica es que utiliza dos líneas para transmitir la información, una para datos y otra para la señal del reloj. Una tercera línea es necesaria para la masa. El nombre que reciben las líneas son: SDA (source data), SCL (source clock) y GND (ground). Posteriormente cuando veamos la IMU se detallarán estas conexiones.

Los dispositivos conectados al bus I²C tienen una dirección única para cada uno y pueden ser *maestros* o *esclavos*. El dispositivo *maestro* es el que inicia la transferencia de datos y además genera la señal de reloj, pero no es necesario que el *maestro* sea siempre el mismo dispositivo, esta característica se la pueden ir pasando los dispositivos que tengan esa capacidad. Esta característica hace que al bus I²C se le denomine bus multimaestro.³

² Más información en http://elinux.org/RPi_Low-level_peripherals

³ Más información en <http://es.wikipedia.org/wiki/I%C2%B2C>

iii. Software

La Raspberry Pi usa mayoritariamente sistemas operativos basados en el núcleo Linux. Raspbian, una distribución derivada de Debian que está optimizada para el hardware de Raspberry Pi, es la distribución utilizada en este proyecto a la que se le han añadido características de tiempo real.

Al mismo tiempo la aplicación del proyecto trabaja sobre una soporte llamado COSME (Control System and Modelling Environment). Es una plataforma software que permite diseñar, construir y ejecutar aplicaciones de control distribuido. COSME nació inicialmente bajo los requerimientos de la empresa TUROMAS, una empresa dedicada al corte de vidrio y el manejo de máquinas por más de 25 años (www.tuomas.com). COSME no es completamente compatible con la norma IEC 61499⁴, a pesar de que utiliza algunos de sus conceptos:

1. Software basado en componentes (FB)
2. Las aplicaciones se crean componiendo redes de FB
3. Permite el diseño y despliegue de sistemas distribuidos (las redes de FB's pueden desplegarse a través de varios controladores)
4. Reconfigurabilidad (los FB's se pueden sustituir o modificar sin parar la aplicación)

Por otra parte, las principales características que lo definen son las siguientes:

⁴ Más información en http://en.wikipedia.org/wiki/IEC_61499

1. Cada instancia tiene un orden de ejecución predefinido, por lo que la aplicación siempre se ejecuta en el mismo orden.

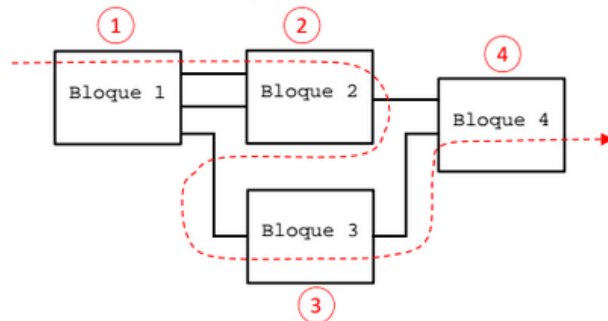


Figura 8: Orden de ejecución de las instancias

2. Método de ejecución. El primer ciclo se ejecutan las funciones inicializa de todas las instancias, luego se ejecutan las funciones normales de cada instancia cada ciclo. Cuando se para COSME se ejecuta la función finaliza de cada instancias.

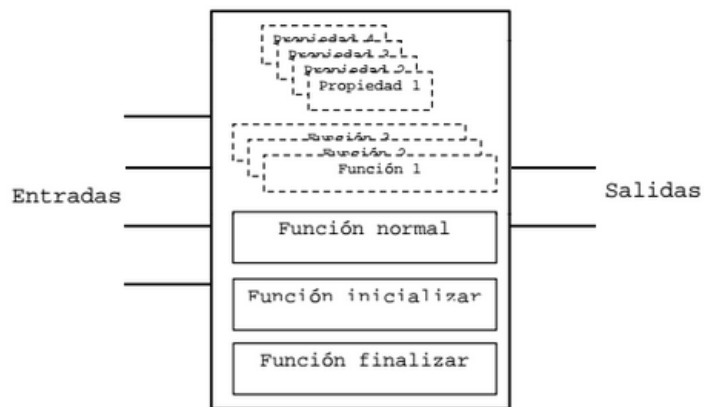


Figura 9: Representación del bloque funcional

iv. Entorno de programación y configuración

Tanto la puesta a punto de la Raspberry Pi como la creación de la aplicación de control se han realizado a través del programa PuTTY. PuTTY es un cliente SSH (Secure Shell) de licencia libre. SSH es el nombre de un protocolo que sirve para acceder a máquinas remotas a través de una red.

Por otro lado para el ajuste del regulador PID y para otras muchas cosas como graficar variables se ha utilizado la aplicación Java MiniBlas.

f. IMU

IMU (Inertial Measurement Unit) o unidad de medición inercial es un dispositivo electrónico que mide e informa acerca de la velocidad, orientación y fuerzas gravitacionales de un aparato, usando una combinación de acelerómetros, giróscopos y magnetómetros. La IMU funciona detectando la actual tasa de aceleración usando un acelerómetro de 3 ejes, detecta los cambios rotacionales como son el yaw pitch y roll mediante giroscopios y mide la orientación en los 3 ejes con el magnetómetro.

En el cuadricóptero se hará uso de la IMU 9 Degrees of Freedom - Sensor Stick de sparkfun que se muestra en la figura 10 y cuyas conexiones para la comunicación I²C son las que se pueden ver en el reverso de la IMU:



Figura 10: Reverso de la IMU

La correcta definición de un sistema de referencia igual para todos los dispositivos de medida es vital, por eso se han definido de la siguiente manera:

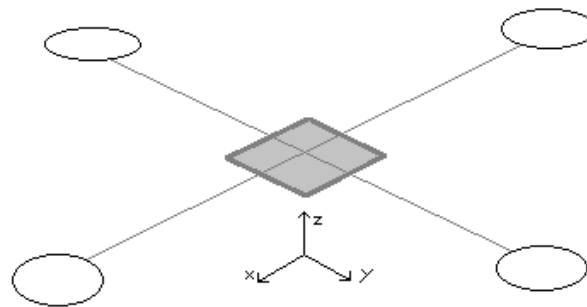


Figura 11: Definición del sistema de referencia

Consta de un acelerómetro ADXL345, de un giroscopio ITG3200 y de un magnetómetro HMC5883L que se explican a continuación.

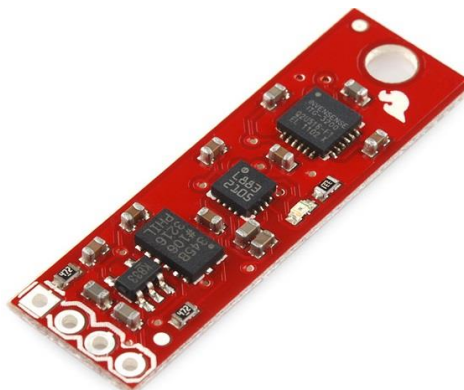


Figura 12: IMU

i. Acelerómetro (ADXL345)

El ADXL345 es un sistema completo de medición de aceleración de 3 ejes del proveedor Analog Devices. Este dispositivo tiene un rango de medición de ± 16 g con una resolución de 13 bits. Permite la comunicación vía SPI pero para este proyecto se ha realizado mediante el bus I²C.

El rango de alimentación va desde 2.0 V a 3.6 V y su consumo es de 23 μ A en modo de medición y desciende a 0.1 μ A en modo de espera. Su alta resolución (3,9 mg / LSB) permite la detección de cambios de inclinación inferiores a 1.0°.

La salida del transductor, son tensiones eléctricas analógicas directamente proporcionales a la aceleración en cada eje. El sistema puede medir aceleración estática (como la gravedad), o aceleración dinámica (como sistemas en movimiento a velocidades no constantes). Las dimensiones del encapsulado son 3 mm \times 5 mm \times 1 mm.⁵

El diagrama funcional de bloques es el siguiente:

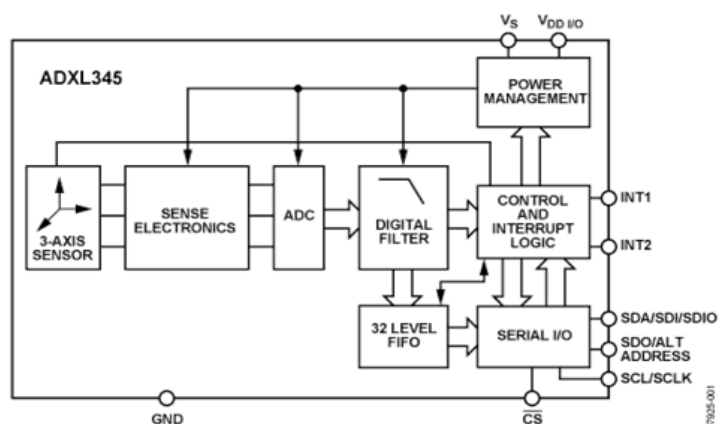


Figura 13: Diagrama de bloques del acelerómetro

⁵ Más información en http://www.analog.com/static/imported-files/data_sheets/ADXL345.pdf

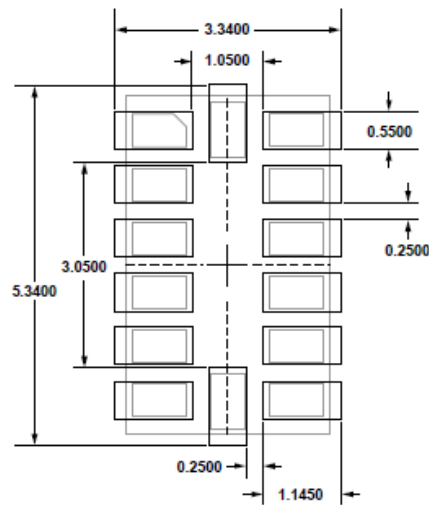


Figura 14: Vista en planta con cotas del acelerómetro

ii. Giróscopo (ITG3200)

El giróscopo es un dispositivo que mide la velocidad angular en un eje sometido a rotación. Es el sensor más importante, ya que sus datos son muy fiables, pero tienen una desventaja frente a otros sensores y es que acumula un error con el tiempo llamado deriva.

El ITG-3200 es un giróscopo con salida digital a través del bus I²C, entregando mediciones correspondientes a los 3 ejes Y, X y Z. Cada medición realizada en su correspondiente eje, es digitalizada por 3 conversores A/D que entregan la información sobre un ancho de 16 bits. Tiene una sensibilidad de 14.375 LSB por °/segundo y un rango de escala completa de $\pm 2,000^\circ / s$ además de poseer un filtro paso bajo programable y un sensor de temperatura interno.

Su tensión de alimentación es muy flexible, va desde 2.1 V hasta los 3.6 V, y su consumo ronda los 6.5 mA en funcionamiento y los 5 μ A en reposo.

Requiere un tiempo de encendido de 50 ms.⁶

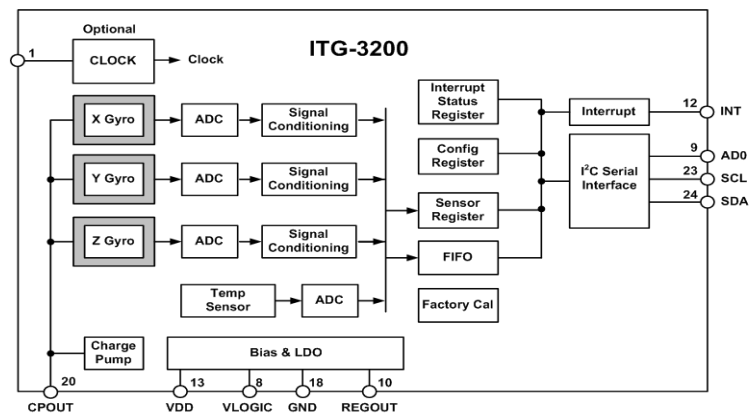


Figura 15: Diagrama de bloques del gir6scopo

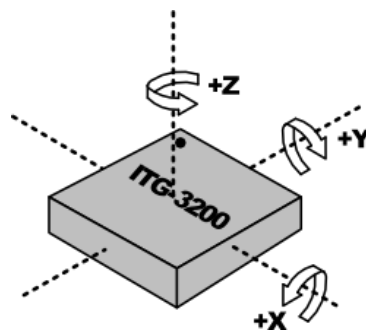


Figura 16: Representaci6n de los ejes del gir6scopo

iii. Magnet6metro (HMC5883L)

Un magnet6metro digital hace la funci6n de una br6jula electr6nica y es capaz de medir la intensidad del campo magn6tico en tres ejes, por lo que el UAV har6 uso de sus mediciones para obtener la orientaci6n.

Los magnet6metros son capaces de medir campos magn6ticos muy d6biles as6 que para garantizar su buen funcionamiento hay que mantenerlo alejado de superficies met6licas e imanes permanentes (motores).

⁶ M6s informaci6n en http://www.analog.com/static/imported-files/data_sheets/ADXL345.pdf

El HMC5883L tiene un ADC con 12 bits de resolución con lo que alcanza el grado de precisión y un rango de medida de ± 8 gauss. La comunicación se realiza vía I²C y su tamaño es considerablemente pequeño (3.0x3.0x0.9 mm). Su alimentación debe estar entre los 2.16 V y 3.6 V y su consumo típico en el modo de medición es de 100 μ A y en el modo de espera el consumo es de 2 μ A.⁷

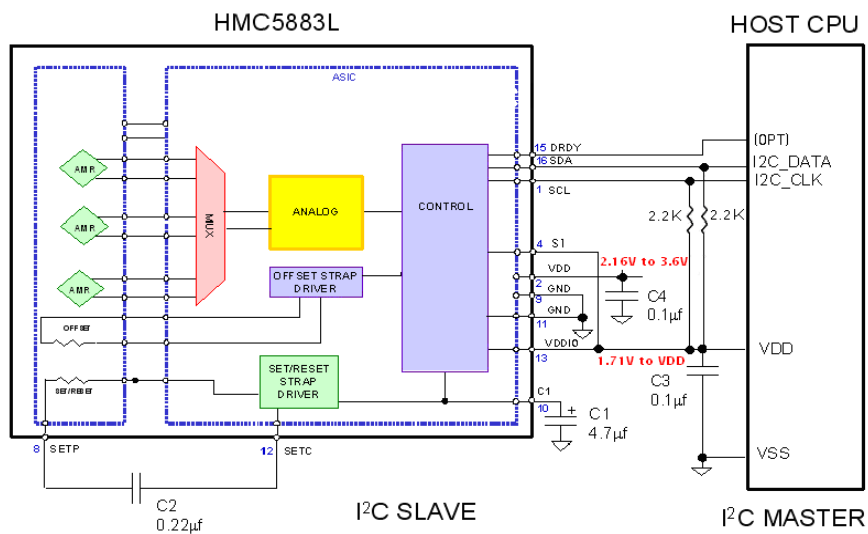


Figura 17: Diagrama de bloques y conexiones del magnetómetro

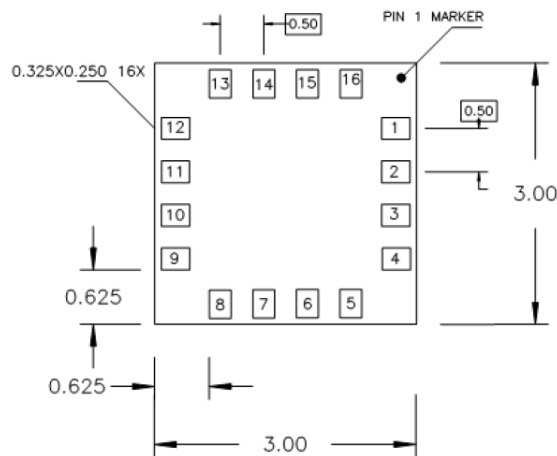


Figura 18: Vista en planta con cotas del magnetómetro

⁷ Información adicional en <https://www.seeedstudio.com/wiki/images/4/42/HMC5883.pdf>

g. Microcontrolador PCA9685

Antes de comenzar con la explicación de este dispositivo es necesario introducir el término PWM para un correcto entendimiento.

El PWM (pulse width modulation) o modulación por anchura de pulsos de una señal es una técnica en la que se modifica el ciclo de trabajo de una señal para controlar la cantidad de energía que se envía a una carga. El ciclo de trabajo (Duty Cycle) se define como un porcentaje del periodo de manera que estando al 50 % la señal PWM estará el mismo tiempo en alto que en bajo.

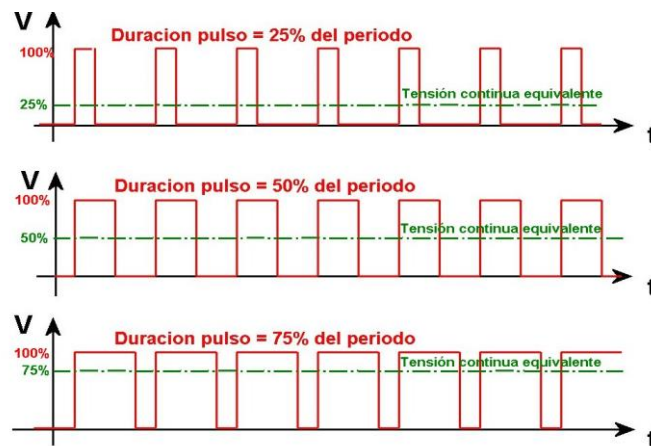


Figura 19: Representación de la señal PWM

El PCA9685 es de vital importancia en el proyecto ya que se encarga de la generación del PWM. Dispone de 16 salidas frente a las 2 salidas que aporta la Raspberry Pi y por otro lado sus 12 bits de resolución frente a los 10 bits de la Raspberry Pi son también una mejora considerable.

La tensión de alimentación puede variar entre 2,3 V y 5,5 V y la corriente máxima necesaria es de 10 mA. La frecuencia del PWM puede oscilar desde 40 Hz a 1000 Hz, o dicho de otra manera, podemos definir un periodo para el PWM entre 25 ms y 1 ms.

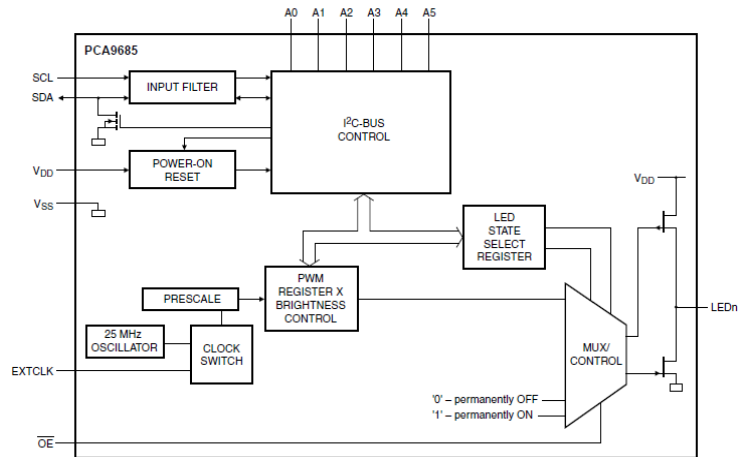


Figura 20: Diagrama de bloques del PCA9685

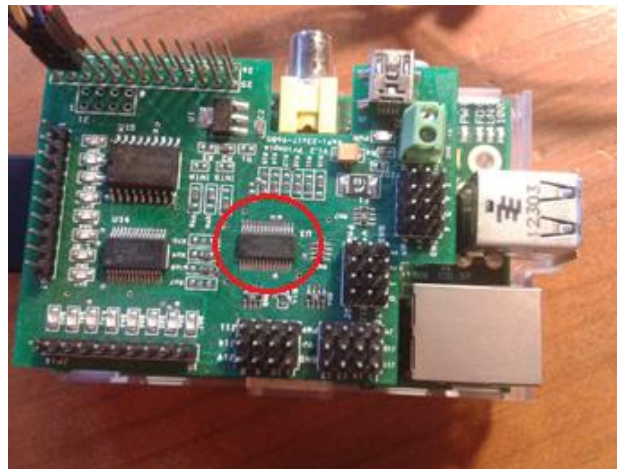


Figura 21: Situación del microcontrolador PCA9685 en la expansion board

h. Actuadores

i. Motores brushless

Un motor eléctrico sin escobillas o motor brushless es un motor eléctrico que no emplea escobillas para realizar el cambio de polaridad en el rotor.

Están compuestos por una parte móvil que es el rotor, donde se encuentran los imanes permanentes, y una parte fija, denominada estator, sobre la cual van enrollados los bobinados de hilo conductor.

En este tipo de motor la corriente eléctrica pasa directamente por los bobinados del estator, por lo tanto aquí no son necesarios ni las escobillas ni el colector que se utilizan en los motores con escobillas (brushed). Esta corriente eléctrica genera un campo electromagnético que interacciona con el campo magnético creado por los imanes permanentes del rotor, haciendo que aparezca una fuerza que hace girar al rotor y por lo tanto al eje del motor.

Un parámetro importante de este tipo de motores es la constante que relaciona la velocidad en RPM del motor sin carga con el valor pico de tensión en las bobinas del motor. A mayores valores para el kv, mayores valores de velocidad, pero menores valores de par y viceversa.

En este caso los motores son de 920 Kv, lo que significa que si a sus bornes les aplicamos 11.1 V la velocidad que alcanzará el motor será de 10212 rpm.

El sentido de giro del motor se determina por las conexiones de las fases.

Los motores elegidos son los Flame 2212 Motor 920Kv de la empresa DJI.



Figura 22: Motor brushless DJI

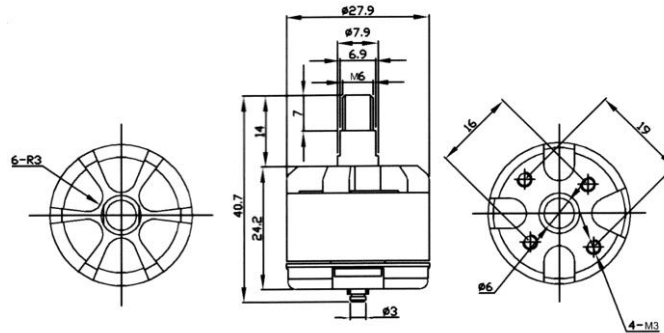


Figura 23: Alzado y planta del motor con cotas

Las características son las siguientes:

- Dimensión: 28X24mm
- Potencia: 920 kv
- Batería: 3S,4S
- Peso: 56 gr
- Corriente estándar: 15-25A , corriente máxima: 30A
- Hélice recomendada: Para una beatería de 3 celdas (11.1 V) el fabricante aconseja una hélice de 10x4.5.

En cuanto a las hélices, es muy importante elegir bien sus dimensiones ya que de ello depende que el cuadricóptero vuele con una buena estabilidad, para este proyecto se ha escogido una hélice de 10x4.5. Las hélices de un cuadricóptero son diseñadas para baja velocidad.

Una forma muy simple de entender el funcionamiento de las hélices es el siguiente: la hélice es como un tornillo contra el aire, va atornillando el aire, es decir, cada vuelta de hélice atornilla un trozo de aire.

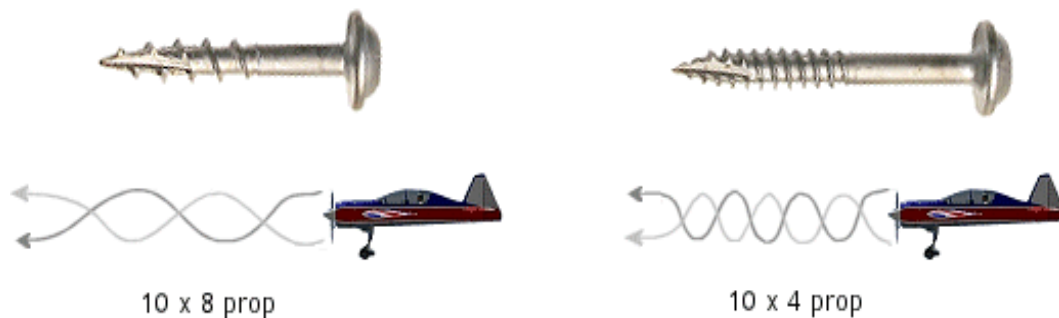


Figura 24: Funcionamiento de la hélice

ii. Variadores de velocidad (ESC)

Un variador de velocidad, en inglés electronic speed control, es un dispositivo, en este caso electrónico, empleado para controlar la velocidad de giro de un motor.

Estas unidades se encargan de generar la secuencia de pulsos trifásicos que son requeridos para provocar el movimiento de los motores.

Los controladores típicos poseen 3 salidas bidireccionales, las cuales generalmente consisten en un conjunto de FETs (transistores que se basan en el campo eléctrico para controlar la conductividad) controlados por un circuito lógico.

Este circuito suele ser un microcontrolador que se encarga de administrar la aceleración y controlar la velocidad de rotación a partir de la señal de control recibida.

Para realizar el proceso anterior, normalmente los ESC tienen que ser configurados con los parámetros del sistema en el cual están trabajando (como por ejemplo el tipo de baterías, las tensiones mínimas de operación segura, etc).

Finalmente poseen las dos conexiones de alimentación que irán conectadas a la batería o fuente de alimentación.

Los ESC elegidos para el cuadricóptero son los DJI Phantom ESC los cuales no requieren previa calibración, vienen con una calibración optimizada de fábrica para los motores DJI.

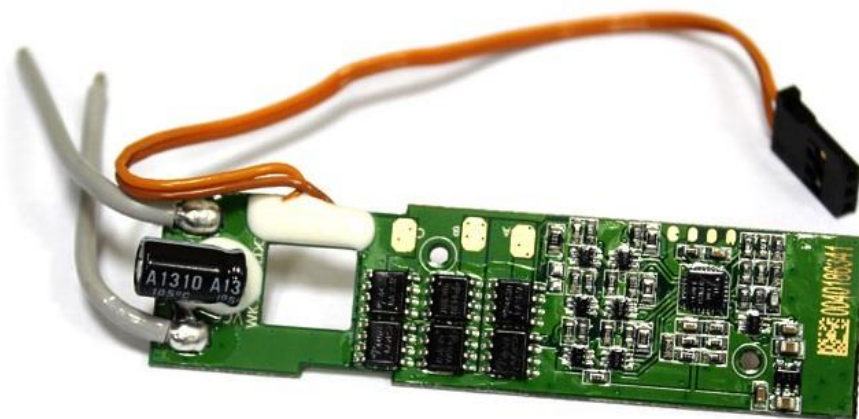


Figura 25: Variador DJI

Sus especificaciones son las siguientes:

- Corriente máxima : 30A
- Frecuencia compatible de la señal de control: 30Hz - 450Hz
- Batería: 3S ~ 4S LiPo

i. Alimentación

La alimentación en un cuadricóptero es un tema delicado, ya que de ello depende entre otras cosas su autonomía.

Las baterías de Polímero de Ión de Litio, también conocidas como baterías de Polímero de Litio o LiPo por su abreviatura en inglés, son un tipo de baterías recargables de alto rendimiento que suelen estar compuestas de varias celdas que se disponen tanto en serie (para aumentar la tensión de salida de la batería) como en paralelo (para aumentar la capacidad y la corriente de descarga de la batería). Este tipo de baterías se caracterizan porque admiten niveles altos de descargas de corriente eléctrica, por lo que son ideales para un cuadricóptero.

La batería elegida es la Turnigy 3.0 cuyas especificaciones son las siguientes:

- Capacidad: 3000mAh
- Configuración: 3S1P / 11.1v / 3Cell
- Constante de descarga: 30C
- Pico de descarga (10sec): 40C
- Peso: 269g

j. Comunicación

La conexión del cuadricóptero con el punto de control se realiza a través del mecanismo de conexión inalámbrica llamado Wi-Fi. Este tipo de comunicación tiene un alcance de unos 20 metros en interiores y algo más en exteriores.

El dispositivo seleccionado es del fabricante Edimax y soporta velocidades de más de 150 Mbps cuando se conecta con modulación 802.11n, la cual es tres veces más rápida que la 802.11g. A diferencia de las otras versiones de Wi-Fi, 802.11n puede trabajar en dos bandas de frecuencias: 2,4 GHz (la que emplean 802.11b y 802.11g) y 5 GHz (la que usa 802.11a).



Figura 26: Dongle Wi-Fi

Este tipo de comunicación tiene una gran ventaja y una gran desventaja. La desventaja es que frente a comunicaciones con radiofrecuencia el alcance está muy limitado, pero existe una gran ventaja y es que con una comunicación Wi-Fi estamos conectados a la red, lo que significa que podemos consultar cualquier variable del cuadricóptero en tiempo real desde cualquier posición siempre y cuando tengamos acceso a red.

k. Procesado de las señales de la IMU

i. Introducción

Debido a la naturaleza analógica de los sensores que se utilizan para percibir el entorno y el estado del cuadricóptero, éstos proporcionan señales de salida que se van a ver sometidas a ciertos niveles de ruido (causado tanto por vibraciones mecánicas como por interferencias electromagnéticas). Por esta razón, resulta de gran importancia acondicionar estas señales de manera que se puedan eliminar los datos provenientes de los sensores que no sean provechosos para el control del vehículo.

ii. Manipulación de los datos

El primer problema que surge al tratar de controlar el cuadricóptero es que unos sensores nos dan la información referida a un sistema de coordenadas y otros nos la dan referida al otro sistema. Estas dos referencias son la referencia suelo (refs) y la referencia cuadricóptero (refc).

El acelerómetro (gravedad), el magnetómetro (Norte Magnético), el barómetro (presión) y el GPS (posición) nos dan la información respecto a la referencia suelo, mientras que el giróscopo y el sensor de ultrasonidos nos la dan referida al cuadricóptero.

Por lo tanto necesitamos recurrir a un sistema de cálculo que nos permita transformar las magnitudes vectoriales de un sistema de referencias al otro y para ello se ha optado por el método de los Cosenos Directores (DCM).

La Matriz de Cosenos Directores (DCM) es una matriz de 3×3 en la que cada fila corresponde a las componentes de los vectores unitarios del sistema referencia suelo en el sistema referencia cuadricóptero, es decir, los cosenos de los ángulos formados por cada eje de la referencia suelo con los ejes de la referencia cuadricóptero.

De manera inversa cada columna corresponde a las componentes de los vectores unitarios del sistema refc en el sistema refs. A la matriz DCM también se le denomina matriz de rotación (R), dado que permite transformar vectores entre dos sistema de referencia que están rotados.

Una de las propiedades interesantes de las matrices de rotación es que permiten agrupar varias rotaciones consecutivas en una única multiplicando las matrices de rotación individuales.

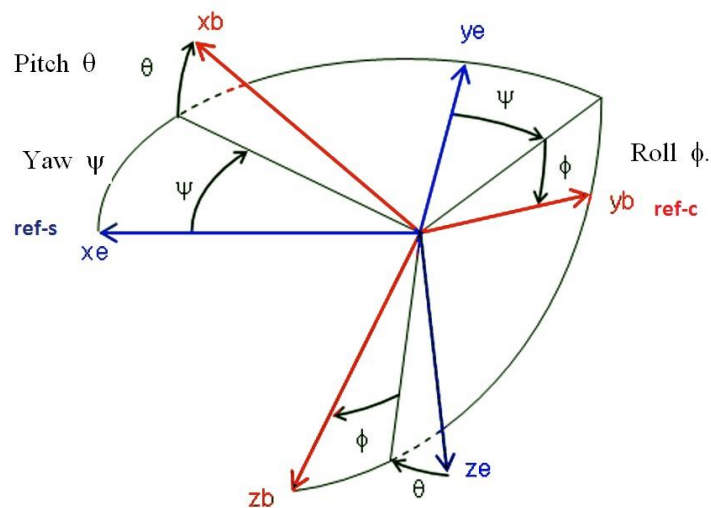


Figura 27: Proyecciones de los vectores de un sistema de referencia (refs) sobre el otro sistema (refc)

$$R = \begin{bmatrix} \cos \theta \cos \Psi & \sin \Phi \sin \theta \cos \Psi - \cos \Phi \sin \Psi & \cos \Phi \sin \theta \cos \Psi + \sin \Phi \sin \Psi \\ \cos \theta \sin \Psi & \sin \Phi \sin \theta \sin \Psi + \cos \Phi \cos \Psi & \cos \Phi \sin \theta \sin \Psi - \sin \Phi \cos \Psi \\ \sin \theta & \sin \Phi \cos \theta & \cos \Phi \cos \theta \end{bmatrix}$$

Ecuación 3: Matriz de rotación

Como los ángulos que puede girar el cuadricóptero en 20 milisegundos son muy pequeños, podemos simplificar algo la matriz de rotación haciendo que todos los cosenos valgan 1 y que los productos seno*seno valgan 0 (figura 28).

Velocidad : 250 ° / s = 5 ° en 20 ms.			
Grados	Radianes	Seno	Coseno
5°	0,0873	0,08715	0,99619
	Seno 5° * Seno 5° =	0,00759	

Figura 28: Comprobación de la simplificación

Suponiendo que durante los 20 milisegundos la velocidad angular (ω) es constante, tenemos que $\Theta = \omega * dt$ con lo que la rotación producida en el instante dt queda reducida a la siguiente matriz llamada matriz de rotación instantánea:

$$R(dt) = \begin{bmatrix} 1 & -d\theta_z & d\theta_y \\ d\theta_z & 1 & -d\theta_x \\ -d\theta_y & d\theta_x & 1 \end{bmatrix}$$

$$\theta_x = \omega_x * dt \quad \theta_y = \omega_y * dt \quad \theta_z = \omega_z * dt$$

Ecuación 4: Matriz de rotación instantánea

Y por la propiedad asociativa de las matrices de rotación tenemos que la matriz de rotación para el instante $(t + dt)$ es igual a la del instante t multiplicada por la matriz de rotación instantánea:

$$R(t + dt) = R(t) \begin{bmatrix} 1 & -d\theta_z & d\theta_y \\ d\theta_z & 1 & -d\theta_x \\ -d\theta_y & d\theta_x & 1 \end{bmatrix}$$

Ecuación 5: Matriz de rotación en el instante $(t + dt)$

Como en el instante inicial los dos sistemas de referencia coinciden la matriz de rotación inicial es:

$$R(0) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Ecuación 6: Matriz de rotación en el instante inicial

Debido a las simplificaciones que hemos efectuado y porque los cálculos no son totalmente exactos poco a poco en la matriz de rotación los vectores unitarios dejan de ser perpendiculares entre sí, los módulos de los vectores dejan de ser exactamente 1, por lo que es necesario corregir esta deriva. El proceso de corrección de estos errores se denomina renormalización. Este método lleva a cabo dos pasos, el primero consiste en conseguir la perpendicularidad entre los vectores y el segundo en hacer los módulos iguales a la unidad.

Primer paso.- Perpendicularidad de los vectores

El producto escalar de 2 vectores es igual a el producto de sus módulos por el coseno del ángulo que forman, por lo tanto el producto escalar de los vectores unitarios X e Y debería ser 0, si esto no es así el resultado nos da una medida de error o falta de perpendicularidad.

Para corregir la desviación se resta a cada vector el producto de la mitad del error por el otro vector:

$$\begin{bmatrix} r_{xx} \\ r_{xy} \\ r_{xz} \end{bmatrix}_{ortogonal} = X_{ortogonal} = X - \frac{error}{2} * Y$$

$$\begin{bmatrix} r_{yx} \\ r_{yy} \\ r_{yz} \end{bmatrix}_{ortogonal} = Y_{ortogonal} = Y - \frac{error}{2} * X$$

Ecuación 7: X e Y ortogonales

Podría hacerse lo mismo comprobando el Z con el X e Y pero existe una solución más sencilla, hacer Z igual al producto vectorial de X e Y.

$$\begin{bmatrix} r_{zx} \\ r_{zy} \\ r_{zz} \end{bmatrix}_{ortogonal} = Z_{ortogonal} = X_{ortogonal} * Y_{ortogonal}$$

Ecuación 8: Z ortogonal

Segundo paso.- Hacer módulos iguales a 1

Una vez conseguidos los vectores ortogonales se hacen los módulos iguales a la unidad de la siguiente manera:

$$X_{normalizado} = \frac{1}{2}(3 - X_{ortogonal} * X_{ortogonal}) * X_{ortogonal}$$

$$Y_{normalizado} = \frac{1}{2}(3 - Y_{ortogonal} * Y_{ortogonal}) * Y_{ortogonal}$$

$$Z_{normalizado} = \frac{1}{2}(3 - Z_{ortogonal} * Z_{ortogonal}) * Z_{ortogonal}$$

Ecuación 9: Vectores normalizados

Es necesario convertir los elementos de la matriz de rotación en ángulos de Euler.

$$\mathbf{R} = \begin{bmatrix} r_{xx} & r_{xy} & r_{xz} \\ r_{yx} & r_{yy} & r_{yz} \\ r_{zx} & r_{zy} & r_{zz} \end{bmatrix} = \begin{bmatrix} \cos \theta \cos \psi & \sin \phi \sin \theta \cos \psi - \cos \phi \sin \psi & \cos \phi \sin \theta \cos \psi + \sin \phi \sin \psi \\ \cos \theta \sin \psi & \sin \phi \sin \theta \sin \psi + \cos \phi \cos \psi & \cos \phi \sin \theta \sin \psi - \sin \phi \cos \psi \\ -\sin \theta & \sin \phi \cos \theta & \cos \phi \cos \theta \end{bmatrix}$$

$$\Phi = \arccos(-r_{zx})$$

$$\Theta = \arctan(r_{zy} / r_{zz})$$

$$\Psi = \arctan(r_{yx} / r_{xx})$$

Figura 29: Obtención de los ángulos de Euler

De esta manera se calculan los ángulos de Euler en grados, que serán entregados posteriormente al control, pero aún hay una cosa más que debemos hacer que es

corregir la deriva o drift que tiene el gir6scopo y esto no se puede pasar por alto ya que sin duda es el sensor m6s relevante a la hora de calcular 6ngulos de vuelo.

La deriva de los ejes X e Y es corregida con el aceler6metro y la deriva en el eje Z con el magnet6metro. Esta correcci6n se basa en un regulador PID que posteriormente es analizado.

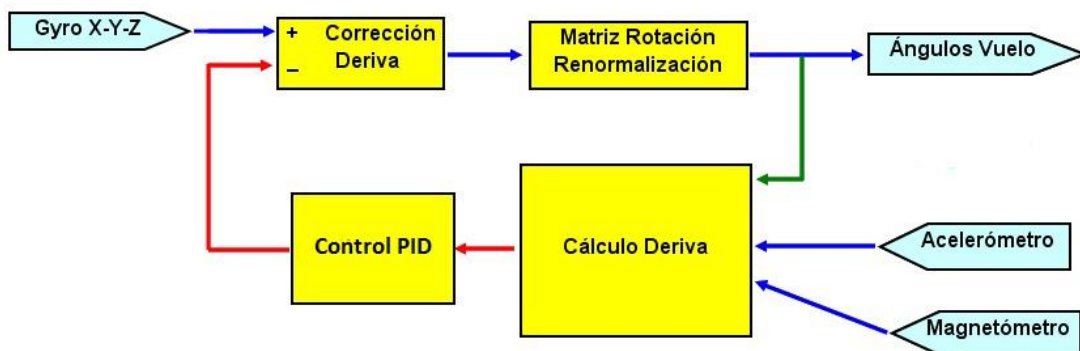


Figura 30: Diagrama de bloques funcional del m6todo

En otras palabras, lo que hace el bucle es:

1. Leer las velocidades angulares del gir6scopo
2. Leer las aceleraciones del aceler6metro
3. Leer los datos del magnet6metro
4. Calcular el error entre los sensores
5. Calcular los factores Proporcional, Integral y Derivativo
6. Corregir las lecturas del gir6scopo con los factores calculados
7. Actualizar la Matriz de Rotaci6n (DCM)
8. Renormalizar DCM
9. Entregar 6ngulos de Vuelo a la regulaci6n

Si se ha producido deriva en el gir6scopo las direcciones del aceler6metro, el cual no tiene deriva, no coincidir6n con las del gir6scopo, por lo que se detectar6 un error que el propio regulador PID corregir6.

k. Control de estabilidad

i. Controlador PID

Los algoritmos de control PID (Proporcional integral derivativo) son los sistemas de control m6s utilizados en la automatizaci6n de procesos debido a su variabilidad y adaptabilidad a diversos procesos controlables.

Estos reguladores funcionan con realimentaci6n, calculando un error entre la entrada deseada y dicha realimentaci6n y disminuyendo este error lo m6ximo posible y lo m6s r6pido posible.

Los algoritmos de control PID poseen 3 componentes cuya suma da como resultado el par6metro de control de la variable manipulada. Estos elementos son: la componente proporcional, la componente integral y la componente derivativa.

El valor proporcional determina la reacci6n del error actual, si este valor es alto estaremos reduciendo el error a costa de aumentar la inestabilidad.

El integral genera una correcci6n proporcional a la integral del error, esto nos asegura que aplicando un esfuerzo de control suficiente, el error de seguimiento se reduce a

cero. Un excesivo factor integral aumentará la sobreamortiguación del sistema y por lo tanto el tiempo de asentamiento.

El factor derivativo determina la variación del error en cada instante de tiempo. Solo afecta mientras halla variación de error, de este modo limita la velocidad a la que se reduce el error impidiendo así que el sistema se vuelva inestable.

La ecuación generalizada de los controladores PID viene dada por la siguiente expresión:

$$u(t) = k_p * e(t) + k_i \int_0^t e(t)dt + k_d \frac{de}{dt}$$

Ecuación 10: Expresión general del regulador PID⁸

Dónde:

- t: corresponde al tiempo transcurrido, o tiempo instantáneo.
- u(t): corresponde a la salida del controlador.
- e(t): corresponde al error (valor deseado – salida real del sistema).
- k_p : corresponde a la ganancia proporcional del controlador.
- k_i : corresponde a la ganancia integral del controlador.
- k_d : corresponde a la ganancia derivativa del controlador.

A continuación se representa gráficamente el diagrama del controlador.

⁸ Información extraída de <http://es.wikipedia.org/wiki/PID>

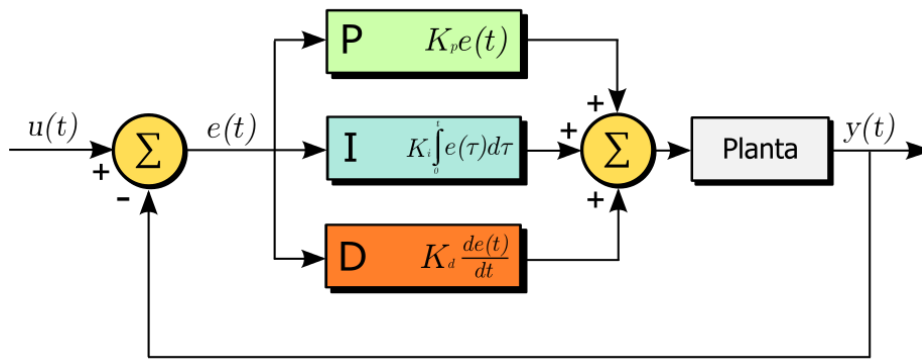


Figura 31: Esquema del regulador PID en paralelo

Para el control de estabilidad se han implementado un regulador para el control de ángulo pitch y otro para el ángulo roll, de los cuales se hablará más adelante.

3. Desarrollo e implementación del COSMEcopter

a. Esquema hardware

A continuación se presenta el esquema hardware del COSMEcopter, en el aparecen todos los elementos necesarios ya mencionados y explicados antes.

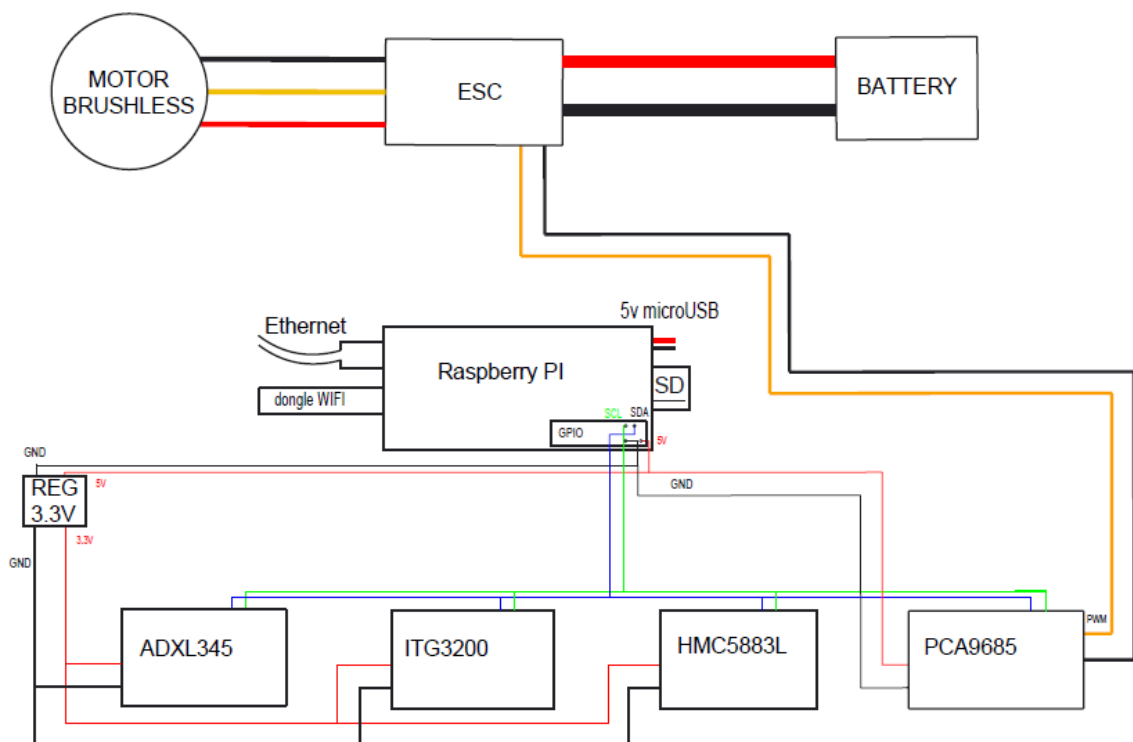


Figura 32: Esquema eléctrico del hardware

El diagrama ha sido esbozado solo para un conjunto motor-variador (figura 32).

b. Aplicación en COSME

La aplicación ha sido diseñada y elaborada sobre la plataforma COSME. De esta manera se puede acceder a cualquier variable del sistema en tiempo real y modificarla, lo que permite ser más eficiente. El diagrama de bloques funcional de la aplicación se presenta a continuación, siendo el orden de ejecución el siguiente:

1. I2C1
2. Acelerómetro1
3. Giróscopo1
4. Magnetómetro1
5. CorrecciónPI_Acc1
6. Promediador1
7. CorrecciónPI_Mag1
8. Actualizar_Matriz1
9. Renormalizar1
10. Conversor_angulos1
11. Referencias1
12. Pid_pitch1
13. Pid_roll1
14. Cambio_magnitud1
15. MicroPWM1

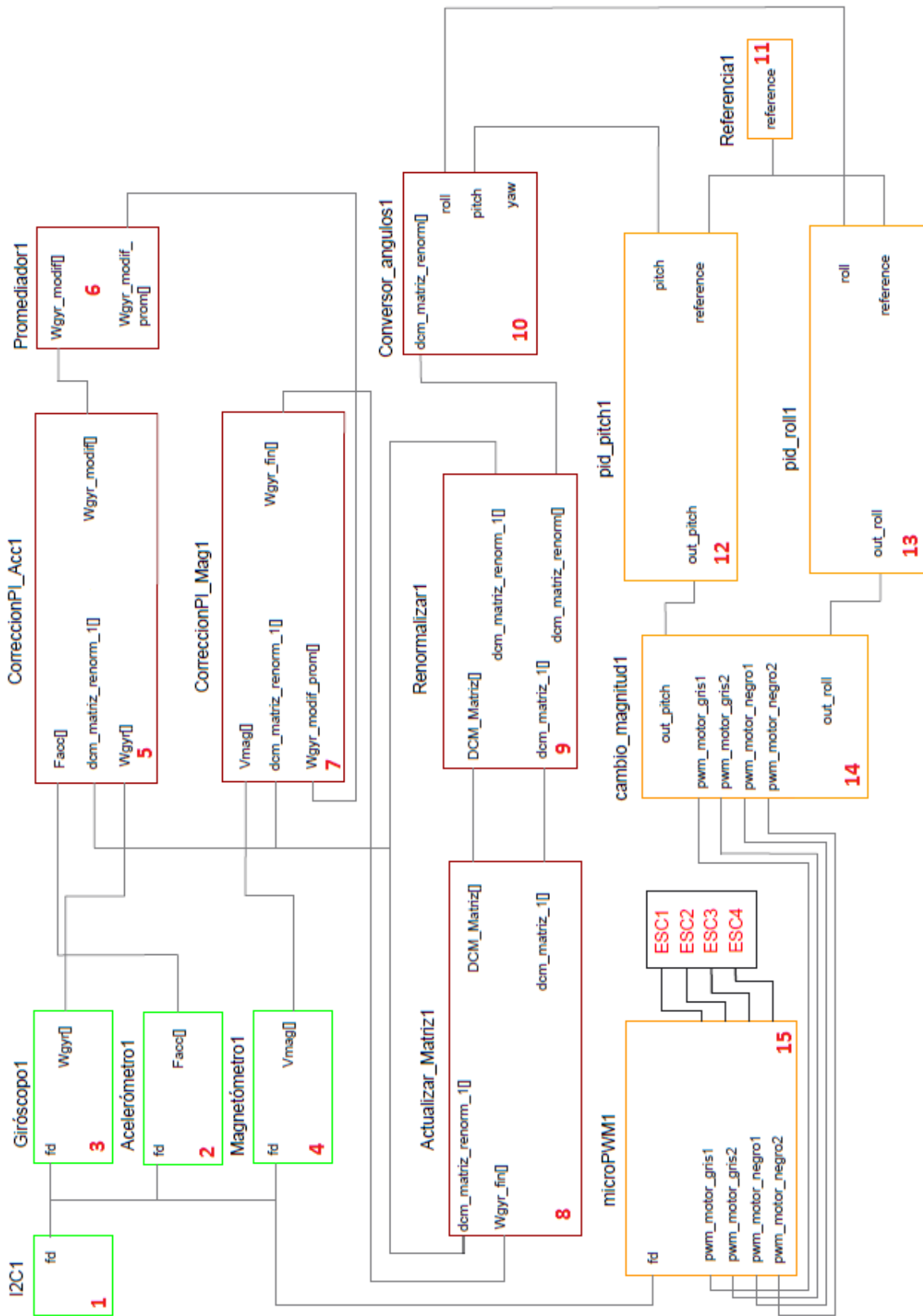


Figura 33: Diagrama de bloques funcionales de la aplicación

Los bloques funcionales en verde están asociados a la adquisición de datos (sensores), los que aparecen en marrón están asociados al tratamiento de dichos datos recogidos y finalmente los bloques en naranja se asocian al control del sistema.

Inicialmente se ejecutan las funciones inicializa de cada instancia en el orden correspondiente. En las instancias referentes a la adquisición de datos se inicializan los sensores con la configuración correspondiente, el giróscopo esta inicializado para que tome medidas cada 1 milisegundo, el magnetómetro cada 13,3 milisegundos entrega el promedio de 8 muestras y el acelerómetro está en modo de medición continua; en el caso de la instancia I²C se inicializa dicho bus de comunicaciones, en la instancia microPWM1 se inicializan los puertos hardware y se define el periodo de la señal PWM de 3 ms ya que es con el periodo que más rango de escala se puede aprovechar. En el resto de instancias se inicializan variables.

En el segundo ciclo ya se han inicializado todas las instancias y comienzan a ejecutarse las funciones normales. En las instancias de los sensores capturamos mediciones de cada uno de ellos. En el tratamiento de datos aparecen dos bloques funcionales llamados CorreccionPI_Acc1 y CorreccionPI_Mag1, el primero consiste en la obtención del error entre los datos del acelerómetro y la matriz de rotación del sistema para sumárselo o restárselo a las mediciones del giróscopo, el segundo consiste en lo mismo pero tan solo es corregida la variable referente al eje Z (yaw). Con estas operaciones obtenemos un vector (X, Y, Z) al que le ha sido eliminada la deriva introducida por el giróscopo.

En la instancia Actualizar_Matriz1 se recibe el vector anterior, se multiplica por el tiempo de ciclo, para pasar de radianes / segundo a radianes, y se almacena en una matriz de 3x3, esta matriz es la que anteriormente definía como matriz de rotación instantánea. Luego se multiplica dicha matriz por la matriz de rotación del sistema, esto nos devuelve la matriz de rotación actualizada.

Posteriormente se pasa a la instancia renormalizar1, cuyo funcionamiento ya ha sido explicado en el apartado del método, del cual se realimenta la matriz de rotación del sistema y finalmente al bloque funcional conversor_angulos1 donde convertiremos estos datos de la matriz de rotación en ángulos de Euler en grados.

Entramos ahora en la etapa de control, donde inicialmente se consulta la referencia, es decir, la referencia es un vector de tres valores (roll, pitch y yaw) que define cómo quiero que este mi sistema. Los reguladores PID intentarán corregir lo antes posible y de la mejor manera la diferencia entre el vector referencia y los ángulos de Euler del cuadricóptero. Dicho regulador se ha implementado de manera no lineal para que su comportamiento sea más agresivo con errores grandes y más suave para errores pequeños. La expresión que gobierna el PID es la siguiente:

$$\begin{aligned} out_PID = & (\alpha + (1 - \alpha) * error) * (kp_PID * error + ki_PID \\ & * integral_PID + kd_PID * derivative_PID) \end{aligned}$$

Ecuación 11: Expresión del PID no lineal

El bloque funcional cambio_magnitud1 consiste en adecuar la señal, que en este caso consiste en pasar de grados a señal PWM. Para ello se ha implementado la siguiente función:

$$pwm = potencia \pm (out_PID * factor_motor)$$

Ecuación 12: Cálculo del pwm en el control de estabilidad

, la variable potencia es un valor entre 1800 y 2800 el cual hace referencia a la potencia a la que vamos a trabajar, es decir, si potencia vale 2300 significa que los motores estarán trabajando al 50%; el \pm será + o será – en función del motor; la variable out_PID es la salida del regulador PID y será la que este variando constantemente; la variable factor_motor es una constante propia de cada motor.

Por último, en la función normal de microPWM1 escribimos el valor del PWM calculado antes en el registro adecuado para que finalmente esta señal llegue a los ESC.

Dado que el tiempo de ejecución de las funciones normales de la aplicación ronda los 10 ms y que atender a las comunicaciones también supone un tiempo se ha optado por un periodo de ejecución de la aplicación de 20 ms. Durante el tiempo sobrante en cada ciclo el sistema esperará.

Finalmente, al cerrar la aplicación se ejecuta la función finaliza de cada una de las instancias donde son puestos a cero los puertos hardware y cerrado el bus I²C.

c. Desarrollo de los bancos de pruebas

Durante el transcurso del proyecto han ido haciendo falta herramientas extra para calibrar o medir ciertas magnitudes. Estas herramientas han sido dos.

La primera es un instrumento mediante el cual conseguimos obtener valores certeros del empuje que realizan cada motor por separado en función de la potencia, es decir, ha sido útil para estimar la gráfica empuje-ciclo PWM.



Figura 34: Instrumento para la medición del empuje

Este instrumento se ha fabricado de madera por su reducido peso frente a metales y por su fácil manejabilidad. Los brazos forman un ángulo de 90° y su funcionamiento es muy simple, el motor a estudiar se coloca en el extremo del brazo vertical de manera que al ejercer un empuje con la hélice esta misma fuerza se vea aplicada en el extremo del brazo horizontal donde hay situada una báscula.

Este método nos permite deshacernos de perturbaciones en las mediciones originadas por el flujo de aire que produce la hélice.

La segunda herramienta consiste en un banco fijo a el cual se le fija un eje del cuadricóptero y el otro queda oscilando libremente en torno al eje fijado. Esto nos sirve de mucho a la hora de ejecutar las primeras pruebas de calibración de los reguladores PID, ya que de este modo es seguro hacer ensayos.

Está fabricado en madera a el cual se le han practicado dos orificios en los que se han alojado dos rodamientos (figura 36) para conseguir que el sistema a la hora de calibrar fuese lo más real posible, es decir, con el menor rozamiento posible.



Figura 35: Banco de calibración



Figura 36: Sistema de agarre con rodamiento

d. Diseño de circuitos impresos

Para el diseño y elaboración de los circuitos se ha utilizado el software libre EAGLE 6.5.0.

i. Placa de alimentación

La distribución de energía en un cuadricóptero es necesaria ya que tenemos una fuente de energía que es la batería y cuatro motores que alimentar, además de la Raspberry Pi, por lo que es necesario elaborar una placa que realice esta función.

El diseño de esta placa es muy sencillo y como lo que interesa de este circuito es que las pistas sean grandes debido a la cantidad de amperios que se descargan, se decidió por el método de elaboración tradicional, es decir, insolando y utilizando disoluciones.

A continuación se muestra el diseño en EAGLE y el circuito terminado.

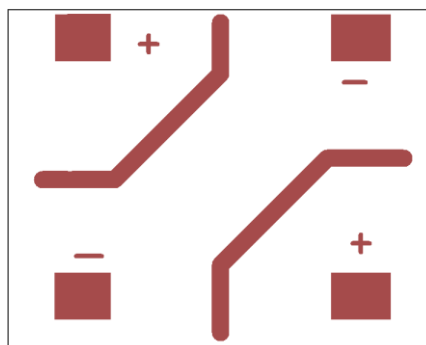


Figura 37: Fichero board de la placa distribuidora de energía

Para la regulación de la tensión de entrada a la Raspberry Pi es necesario un regulador, en este caso se ha seleccionado el regulador de la serie 7805 (figura 38) debido a que

la tensión de entrada permitida es de hasta 25 V (nuestra batería es de 11.1 V) y a que a la salida del regulador las condiciones son perfectas, 5 V y 1.5 A máximo.⁹

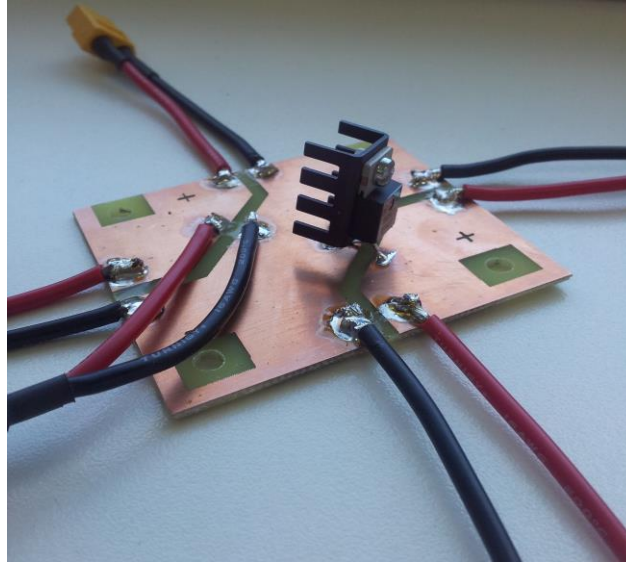


Figura 38: Circuito de distribución de energía

ii. Circuito impreso de sensado

Al comienzo del proyecto se pensó en realizar el circuito impreso que contuviese los tres sensores acelerómetro, giróscopo y magnetómetro, con la intención de abaratar el coste del proyecto.

⁹ Información extraída de [www.mccsemi.com/up_pdf/MC7805CT\(TO-220\).pdf](http://www.mccsemi.com/up_pdf/MC7805CT(TO-220).pdf)

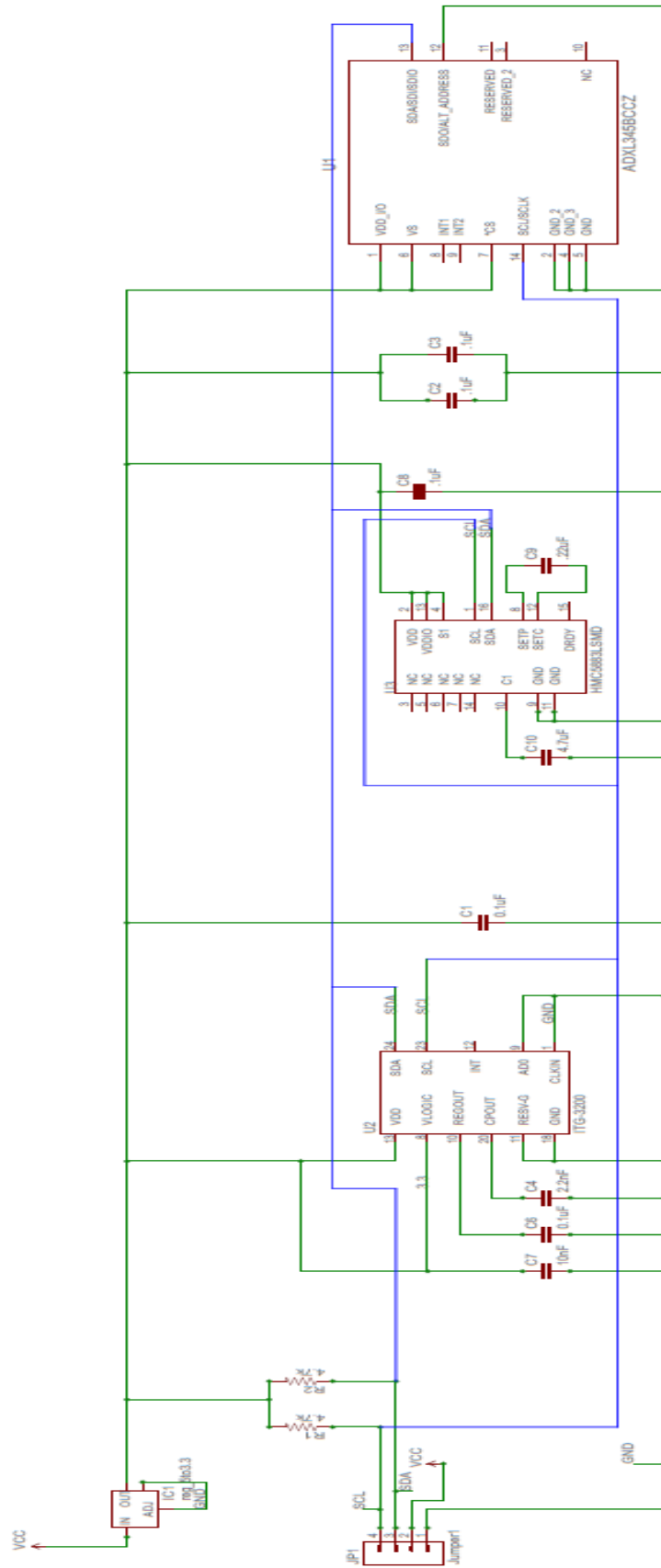


Figura 39: Fichero schematic del circuito de sensado

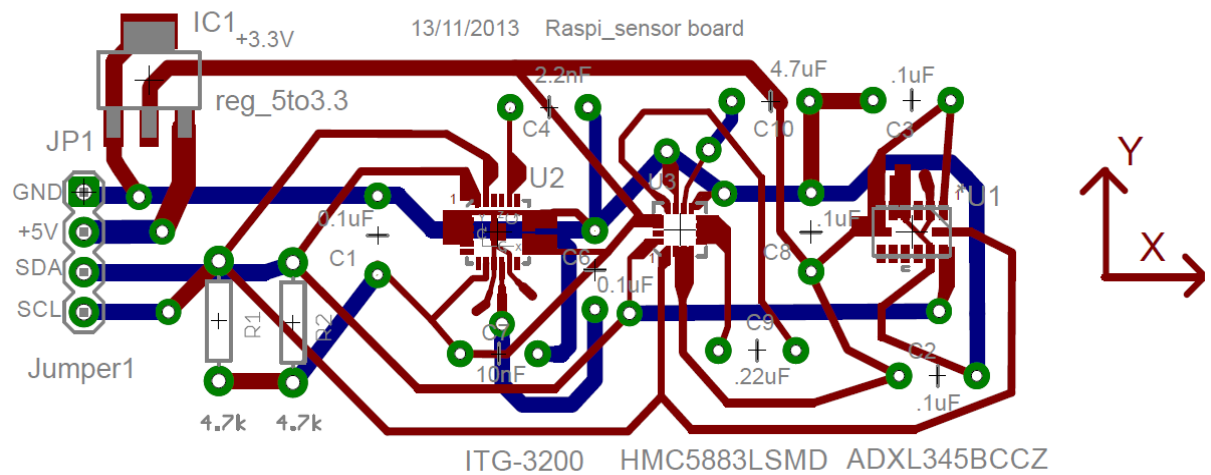


Figura 40: Fichero board del circuito de sensado

Finalmente generamos los ficheros gerber para exportarlos a la fresadora de circuitos impresos, la cual nos devuelve un muy buen resultado (figura 41).

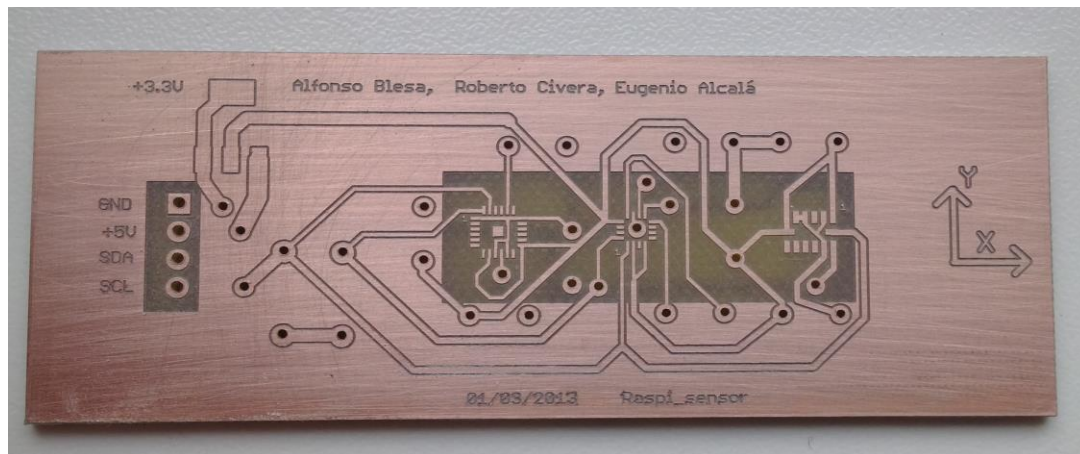


Figura 41: Circuito de sensado sin componentes

Una vez terminada se procede a la etapa de soldar componentes, es decir, la tarea complicada. Las empresas que se dedican a la fabricación de PCB's cuentan con el método llamado soldadura por ola, el cual favorece y facilita la tarea para

componentes muy pequeños como es nuestro caso, pero esa tecnología está fuera de nuestro alcance por lo que se intentó soldar los encapsulados de la forma tradicional obteniendo como resultado el siguiente:

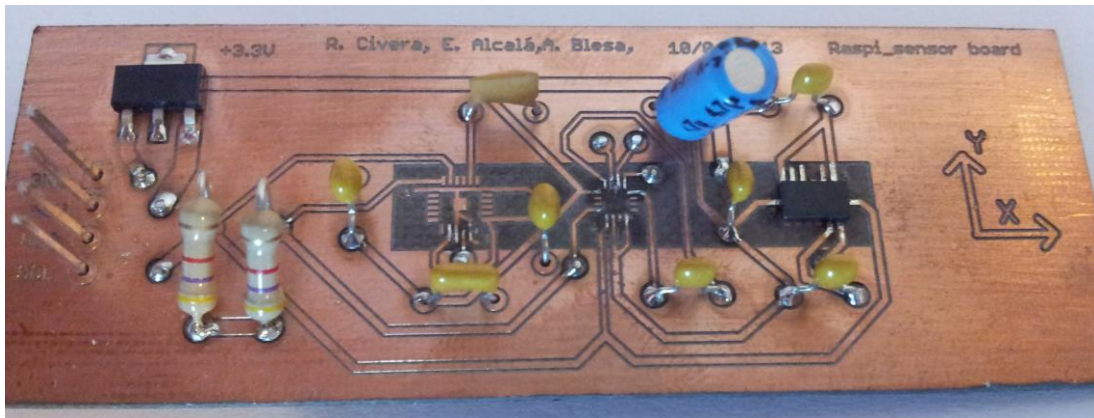


Figura 42: Circuito de sensado con componentes soldados

Como se puede apreciar en la figura 42 la placa no ha podido ser realizada debido a la alta complejidad de la soldadura.

e. Pruebas

i. Puesta en marcha y calibración de los sensores

Poner en correcto funcionamiento el conjunto de los tres sensores no ha resultado ser tarea fácil, todos ellos han sido probados en todos sus modos de configuración escogiendo como resultado final el mejor de ellos.

El acelerómetro ADXL345 ha sido configurado para el modo de medición continua, con una resolución de 10 bits y un rango de medición de -2 g a +2 g.¹⁰

El giróscopo ITG3200 está configurado para que realice mediciones cada 1 ms, es decir, con una frecuencia de muestreo de 1000 Hz y tiene un rango de $\pm 2000 \frac{\circ}{s}$, pero este sensor también posee un filtro paso bajo regulable por software el cual finalmente se ha establecido en 5 Hz, esto quiere decir que movimientos bruscos y rápidos los despreciará.¹¹

El magnetómetro se ha modificado para que entregue la media de 8 muestras cada 13 ms midiendo en el rango de los ± 1.3 Ga.¹²

Inicialmente se optó por el método más sencillo de implementar que es el llamado filtro complementario. Este método lo que hace es dar mayor o menor peso a los valores provenientes del giróscopo y del acelerómetro para conseguir un valor más libre de ruidos y con menor error que el que entrega el giróscopo. La ecuación es la siguiente:

$$angulo_{pitch} = k_{giroscopo} * giroscopo_{pitch} + k_{acelerómetro} * acelerómetro_{pitch}$$

$$angulo_{roll} = k_{giroscopo} * giroscopo_{roll} + k_{acelerómetro} * acelerómetro_{roll}$$

Ecuación 13: Expresiones para el cálculo de los ángulos de vuelo

La constante $k_{giroscopo}$ suele estar comprendida entre 0.93 y 0.98 y la constante $k_{acelerómetro}$ entre 0.07 y 0.02 ya que la suma ha de ser igual a la unidad.

¹⁰ Información extraída de http://www.analog.com/static/imported-files/data_sheets/ADXL345.pdf

¹¹ Información extraída de <https://www.sparkfun.com/datasheets/Sensors/Gyro/PS-ITG-3200-00-01.4.pdf>

¹² Información extraída de <https://www.seeedstudio.com/wiki/images/4/42/HMC5883.pdf>

Realizamos una prueba que consistía en girar a 90° y luego a -90° el ángulo roll y luego a -90° el ángulo pitch.

Tras una serie de pruebas cuyo resultado no nos servía llegamos al mejor resultado, para $k_{giroscopo} = 0.95$ el resultado era el siguiente:

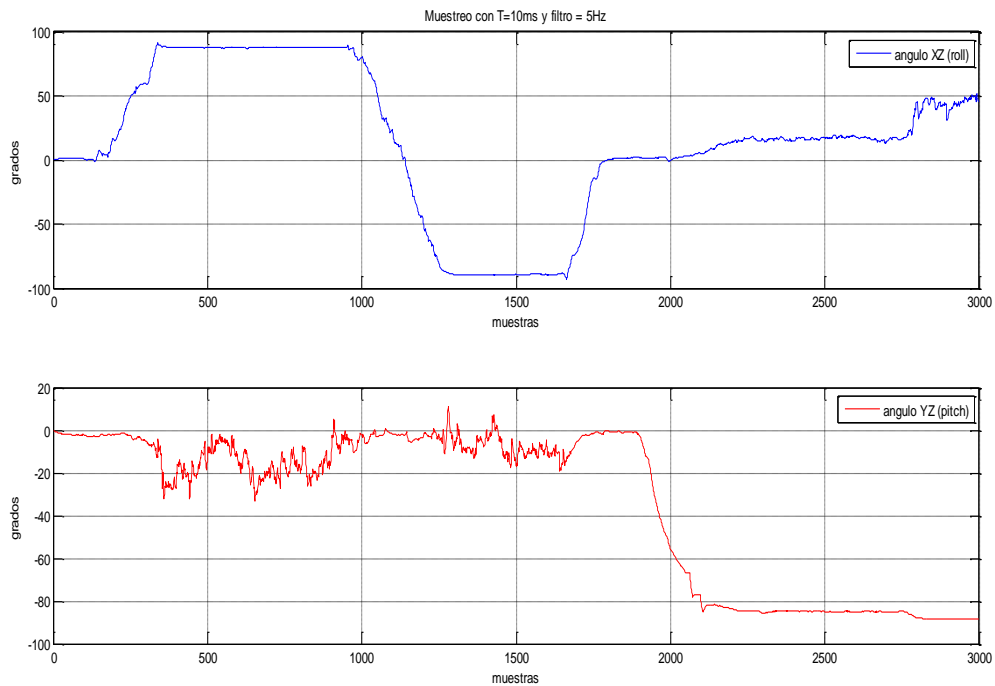


Figura 43: Representación ángulos roll y pitch en Matlab.

Se puede apreciar que al cabo de una serie de muestras el ángulo roll ya no permanece en 0 grados, esto es debido a que damos mucho peso a la variable que aporta error con el tiempo. El ruido que aparece es introducido por el acelerómetro.

Debido a esto me veo obligado a buscar otro método, otro algoritmo para el tratamiento de estos datos. Este nuevo método es la matriz de cosenos directores (DCM) explicado anteriormente.

Como ya hemos visto, el algoritmo es relativamente más complejo pero los resultados también son más satisfactorios. Una prueba de ello es la siguiente gráfica en COSME MiniBlas en la que muevo el cuadricóptero de 90° a -90° en los ejes X e Y:

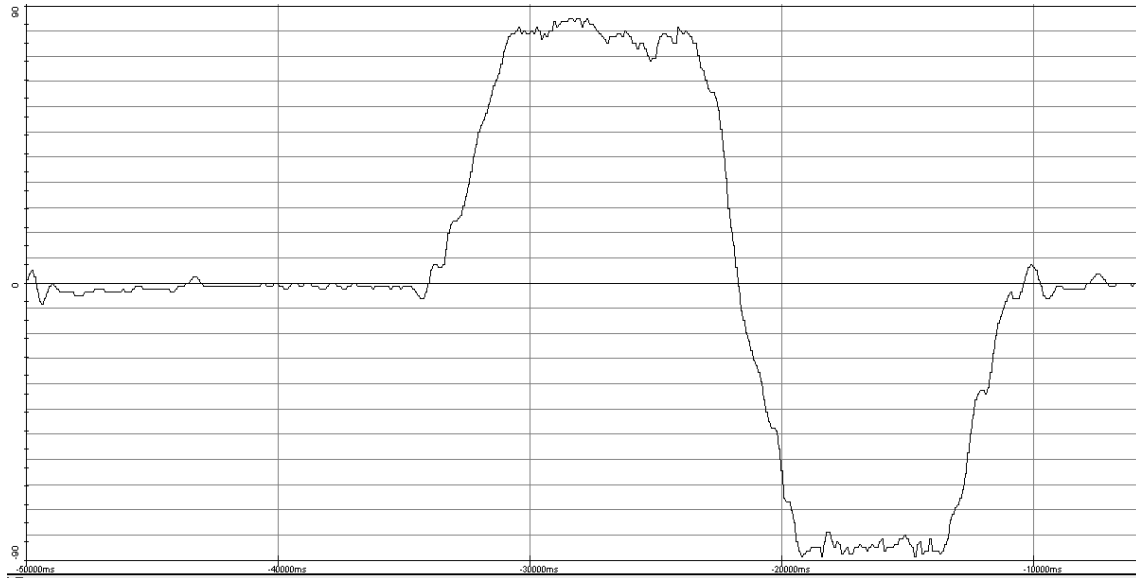


Figura 44: Ángulo pitch en MiniBlas.

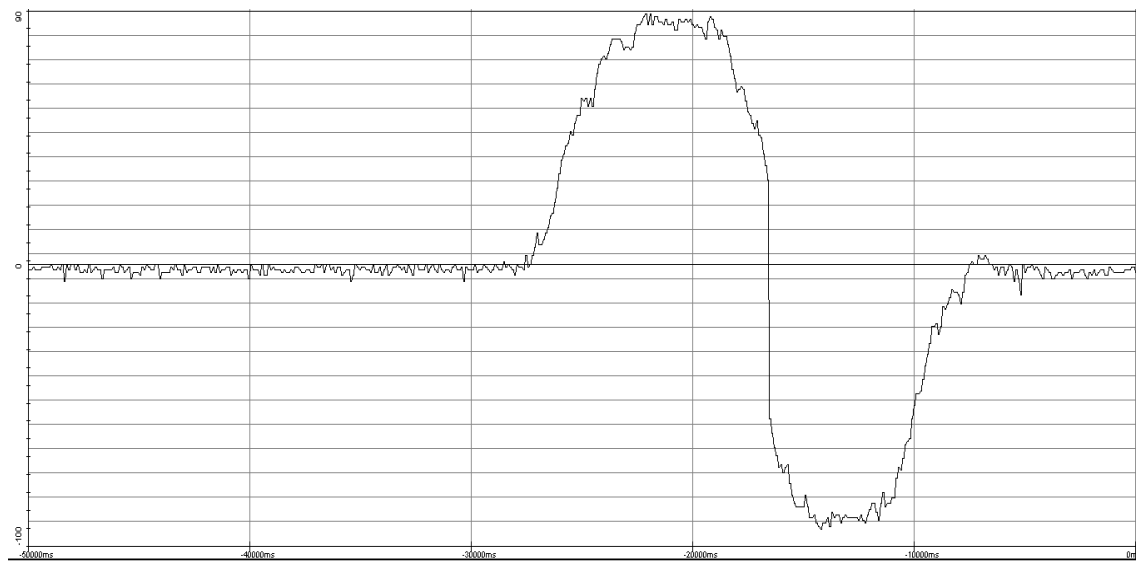


Figura 45: Ángulo roll en MiniBlas.

ii. Calibración de los 4 motores. Curvas de potencia

Para la puesta a punto de los motores lo primero es ver cómo se comporta cada motor por separado, para ello se ha hecho uso del instrumento fabricado.

Una vez montado el sistema de medición se procede a elaborar una tabla para cada motor que muestre el porcentaje del ciclo PWM frente al empuje en gramos. Estos datos son introducidos en un script en Matlab el cual nos devolverá las gráficas de los cuatro motores.

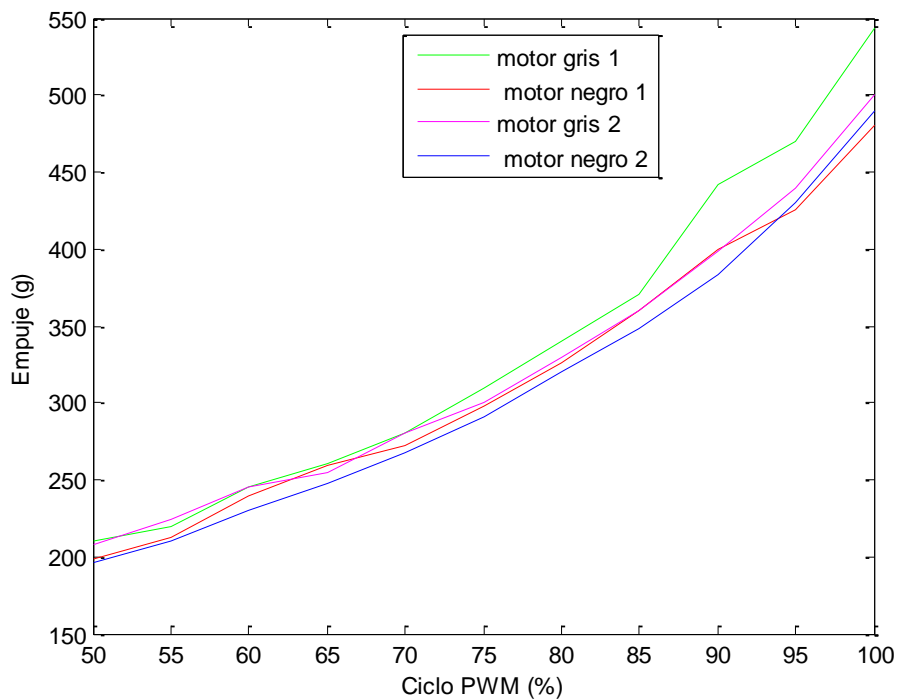


Figura 46: Gráfica empuje frente a PWM

El objetivo de esta calibración es ajustar las parábolas de los motores a una parábola igual para todos con el fin de facilitar posteriormente el control.

Las parábolas se han corregido en función de la recta del motor negro 2 (parábola azul), de esta manera tan solo se ha tenido que aplicar un factor de corrección a cada curva y el resultado es el siguiente:

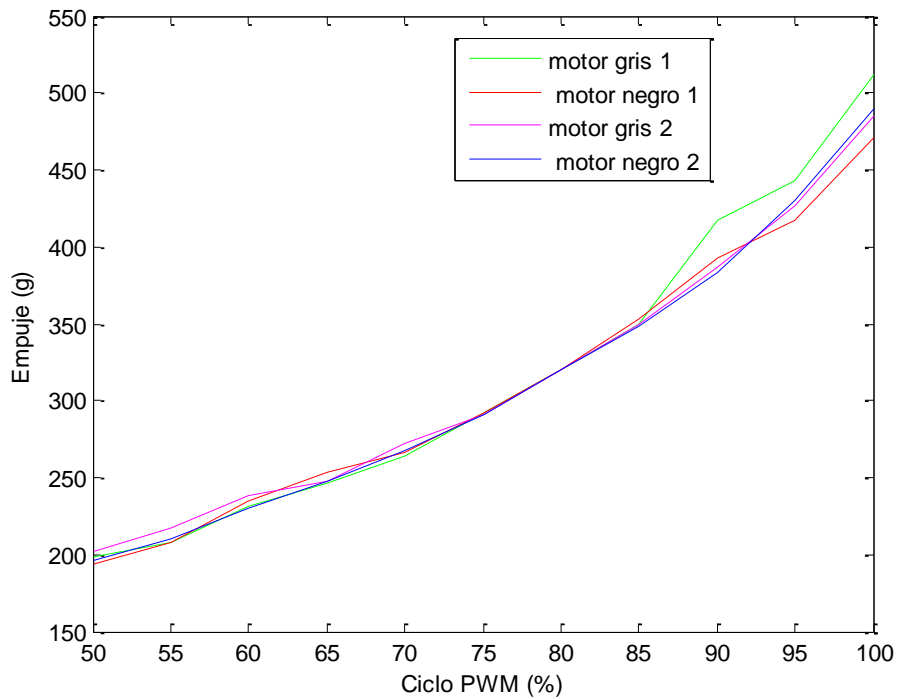


Figura 47: Grafica empuje frente a PWM aplicado un factor de corrección a la curva de cada motor

El tramo mejor ajustado estará en el rango del 73 % al 83 % del ciclo PWM lo cual es correcto ya que posteriormente el UAV trabajará en ese rango de potencia.

iii. Calibración del control PID

La puesta a punto del control de estabilidad requiere dos reguladores PID, es decir, uno por cada eje, por lo que a la hora de ajustar los parámetros del regulador se han tratado por separado. Para esta calibración se ha utilizado el banco de pruebas comentado en el apartado 3.c (figura 35).

El método seguido para calibrar el controlador es el de Ziegler-Nichols en lazo cerrado, el cual permite ajustar o "sintonizar" un regulador PID de forma empírica, sin necesidad de conocer las ecuaciones de la planta. Estas reglas de ajuste propuestas por Ziegler y Nichols fueron publicadas en 1942 y desde entonces es uno de los métodos de sintonización más difundido y utilizado.

Este método consiste en reducir al mínimo la acción derivativa y la acción integral del regulador PID y aumentar poco a poco la ganancia proporcional hasta que el sistema oscile de forma mantenida ante cualquier perturbación. Esta oscilación debe ser lineal, sin saturaciones (figura 48). En este momento hay que medir la ganancia proporcional, llamada ganancia crítica o K_c , y el periodo de oscilación T_c en segundos.

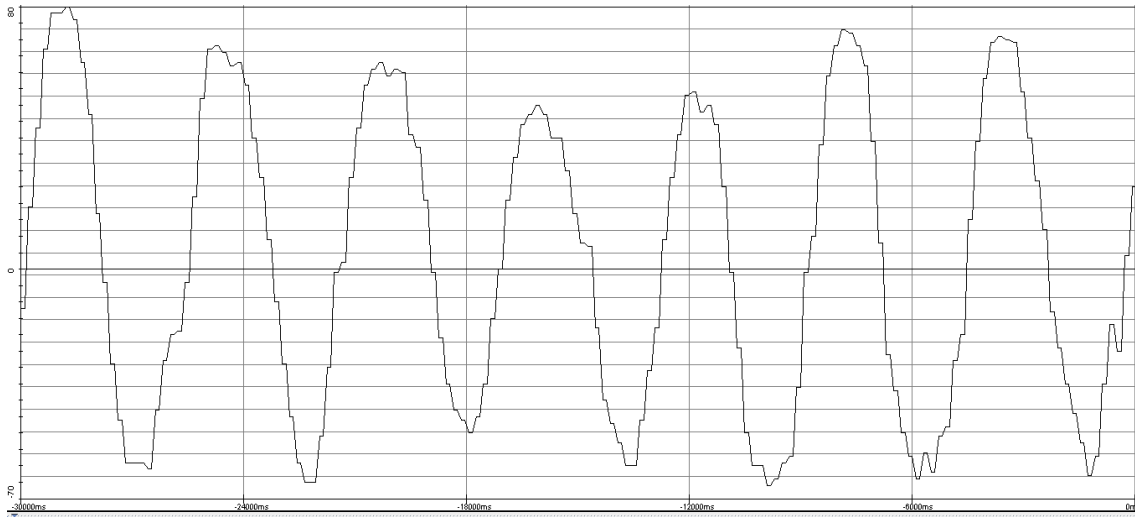


Figura 48: Sistema críticamente estable en miniBlas

Una vez hallados estos dos parámetros se pueden calcular los parámetros del controlador PID de la siguiente manera:

	Kp	Ti	Td
P	$0.50 \cdot K_c$		
PI	$0.45 \cdot K_c$	$0.83 \cdot T_c$	
PID	$0.59 \cdot K_c$	$0.50 \cdot T_c$	$0.125 \cdot T_c$

Tabla 2: Tabla de Ziegler-Nichols

Si los valores de tiempo T_i y T_d se traducen a ganancias, se obtiene que:

$$K_i = K_p / T_i \quad \text{y} \quad K_d = K_p * T_d$$

Ecuación 14: Relación entre tiempos integral y derivativo y la constante proporcional

Siguiendo este método para el cuadricóptero conseguimos que el sistema sea “críticamente estable” con $K_c = 0.7$ y un periodo de oscilación $T_c = 9$ s en el eje X. Con estos datos obtenemos que $K_p = 0.525$, $K_i = 0.0933$ y $K_d = 0.4725$. Posteriormente el cuadricóptero se reorienta para poder calibrar ahora el eje Y ya que el cuadricóptero no tiene el mismo peso sobre un eje que otro.¹³

Pero con estas constantes el sistema no está totalmente regulado, es decir, son orientativas, ahora hay que ajustarlo del todo mediante prueba y error.

iv. Consumos del cuadricóptero

El estudio del consumo del UAV es importante para estimar el tiempo de vuelo. En esta prueba lo que se ha hecho es medir los consumos en función del porcentaje PWM aplicado en un motor y posteriormente hallar el consumo total del cuadricóptero.

La siguiente tabla muestra los resultados.

¹³ Información extraída de <https://sites.google.com/site/picuino/ziegler-nichols>

Consumos del cuadricoptero			
Ciclo PWM (%)	Consumo un motor (A)	Potencia un motor (W)	Potencia los 4 motores (W)
20	1,08	11,988	47,952
30	1,54	17,094	68,376
40	2,02	22,422	89,688
50	2,5	27,75	111
60	3	33,3	133,2
70	3,45	38,295	153,18
80	4,1	45,51	182,04
90	5,18	57,498	229,992
100	6,5	72,15	288,6

Tabla 3: Tabla de consumos y potencias

Con estos datos podemos estimar la autonomía de vuelo de la siguiente manera:

$$\frac{\text{Amperaje batería} * 60}{\text{Consumo en amperios}} = \text{tiempo de vuelo en minutos}$$

Entonces si el amperaje de la batería es de 3000mAh y sabiendo el consumo en amperios para diferentes rangos de potencia podemos estimar el tiempo de vuelo.

Ciclo PWM (%)	Consumo 4 motores (A)	Autonomía (min)
20	4,32	42
30	6,16	29
40	8,08	22
50	10	18
60	12	15
70	13,8	13
80	16,4	11
90	20,72	9
100	26	7

Tabla 4: Cálculo de la autonomía

Dado que el cuadricóptero trabaja en el rango entre el 70% y el 80% la autonomía será de entre 13 y 11 minutos.

4. Componentes y coste

La siguiente tabla muestra los componentes empleados en el proyecto con sus respectivos costes y proveedores. Los elementos fabricados y piezas tales como cables, tornillos, etc, están incluidas en otros.

Material	Cantidad	Coste unidad(€)	Coste total (€)	Proveedor
Aluminio	1 metro	6	6	Bauhaus
Poliestireno	180 cm2	3	3	Bauhaus
Bateria	2	22	44	hobbyking
Balanceador	1	28	28	hobbyking
Raspberry Pi + dongle	1	40	40	element14
PCA9685	1	25	25	pridopia
3 sensores	1	100	100	SparkFun
ESC's	4	23	92	robbe
Motores	4	30	120	robbe
Hélices	4	13	52	robbe
Otros	30	1	30	
TOTAL EUROS			540	

Tabla 5: Lista de componentes con sus respectivos costes

5. Conclusiones y futuro

a. Conclusiones

- La configuración y puesta en marcha de los sensores ha resultado ser más complicada de lo previsto ya que al no trabajar con una plataforma como por ejemplo Arduino no existen librerías predefinidas, aunque por otro lado resulto ser esencial para el correcto entendimiento y para poder mejorar la configuración después.
- El diseño y la elaboración del circuito impreso de sensado nos costó un tiempo muy importante y finalmente no se pudo llevar a cabo, pero sirvió para entrar en contacto con el diseño de circuitos impresos.
- La estructura podría haberse construido algo más pequeña con el fin de reducir los momentos generados en la estructura.
- El proceso de pruebas de los conjuntos motor-hélice resultó ser bastante peligroso ya que los motores giran a muy altas velocidades y existe el riesgo de que la programación sea errónea. Las pruebas se realizaron a cierta distancia y con todos los mecanismos debidamente anclados.
- La integración de todas las tecnologías desarrolladas resultó llevar más tiempo del establecido inicialmente.
- La regulación y puesta a punto del control de estabilidad resultó no ser trivial y acabó llevando más tiempo del deseado.

- La Raspberry Pi es una buena plataforma para este proyecto debido a que tiene capacidad de procesamiento suficiente y trabaja en tiempo real.
- Las hélices no han sido calibradas por lo que podrían ser una fuente de error a pesar de la fiabilidad que te da el fabricante.
- El desarrollo de un proyecto como el presente, requiere de mucha dedicación, planificación y sobre todo tiempo debido a la gran cantidad de investigación, pruebas y experimentos que se deben realizar en tan poco tiempo.

b. Trabajo futuro

- Realizar un control de altitud robusto uniendo a través de un algoritmo las mediciones del acelerómetro y un barómetro.
- Manejar el cuadricóptero con un Smartphone.
- Proveer al cuadricóptero de sensores de ultrasonidos para interactuar con el entorno que le rodea.
- Incluir la cámara Raspberry Pi para poder hacer fotos y poder ver en tiempo real a través de un dispositivo por donde se desplaza.
- Reducir el peso total del cuadricóptero y proteger más los dispositivos electrónicos.

6. Agradecimientos

El presente trabajo ha sido realizado bajo la supervisión de Alfonso Blesa Gascón como director y de Carlos Catalán Cantero y Félix Serna Fortea como parte del tribunal, a quienes me gustaría expresar mi agradecimiento por su tiempo y dedicación para que este proyecto saliese adelante, sin ellos no hubiese sido posible.

A Miguel Ángel García Monclús por su paciencia, tiempo y dedicación en las tareas mecánicas del proyecto y a Alberto Azuara García por su gran ayuda en las tareas de software.

A Paula Vilardell Maymus y a mi familia por su gran apoyo día a día, ellos han sido quienes me han ayudado a ver las cosas de otra manera cuando más lo requería.

BIBLIOGRAFÍA

Páginas Web

1. "Raspberry Pi"

http://es.wikipedia.org/wiki/Raspberry_Pi

2. "Explicación del motor brushless"

<http://www.cochesrc.com/motor-electrico-brushless-funcionamiento-y-caracteristicas-a3607.html>

3. "Explicación del variador de velocidad (ESC)"

<http://tallerdedalo.es/web/ESC>

4. "Matriz de cosenos directores (DCM)"

<https://sites.google.com/site/kuadricoptero/home/4-dcm/4-01-dcm-01>

5. "Puesta en marcha giróscopo, acelerómetro y magnetómetro"

<https://sites.google.com/site/mikuadricoptero/home/download/2-62-gyro-3>

6. "Fuerza de sustentación"

<http://es.wikipedia.org/wiki/Sustentaci%C3%B3n>

7. "Orientación IMU"

<http://bildr.org/2012/03/stable-orientation-digital-imu-6dof-arduino/>

8. "Control PID"

<http://es.wikipedia.org/wiki/PID>

9. "Hoja de características del microcontrolador PCA9685"

<http://www.adafruit.com/datasheets/PCA9685.pdf>

10. "Hoja de características del giróscopo ITG3200"

<https://www.sparkfun.com/datasheets/Sensors/Gyro/PS-ITG-3200-00-01.4.pdf>

11. "Hoja de características del acelerómetro ADXL345"

http://www.analog.com/static/imported-files/data_sheets/ADXL345.pdf

12. "Hoja de características del magnetómetro HMC5883"

<https://www.seeedstudio.com/wiki/images/4/42/HMC5883.pdf>

13. "Hoja de características del regulador 7805"

[http://www.mccsemi.com/up_pdf/MC7805CT\(TO-220\).pdf](http://www.mccsemi.com/up_pdf/MC7805CT(TO-220).pdf)

14. "Método de Ziegler-Nichols"

<https://sites.google.com/site/picuino/ziegler-nichols>

15. "Hélices"

<http://cuadricoptero.net/helices-para-cuadricoptero/>

ANEXOS

Anexo A: Galería fotográfica

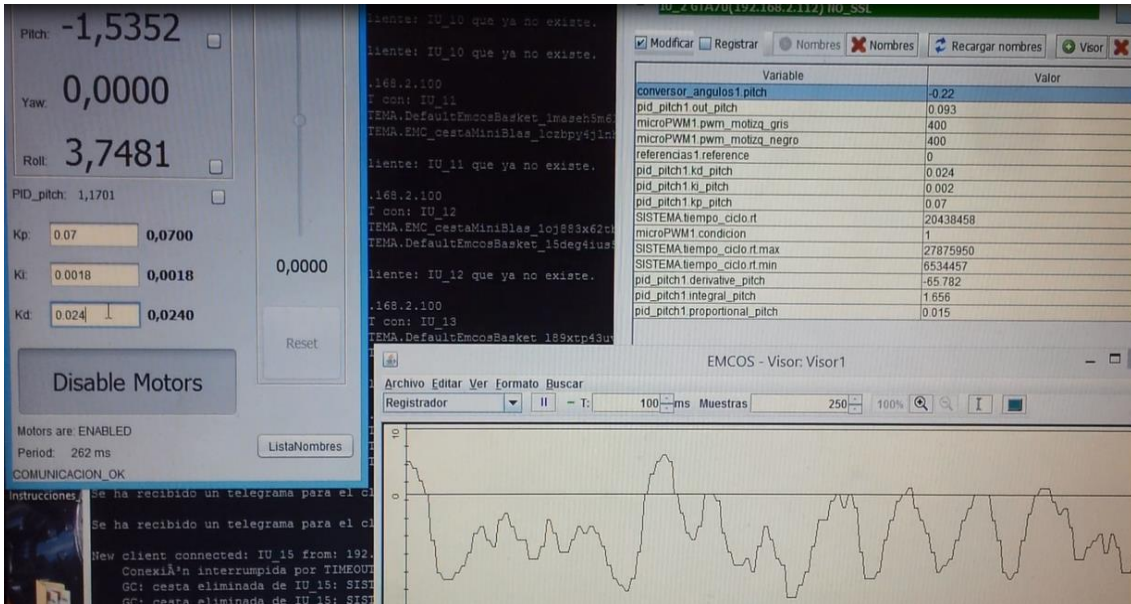


Ilustración 1: Trabajando con miniBlas y la aplicación Java

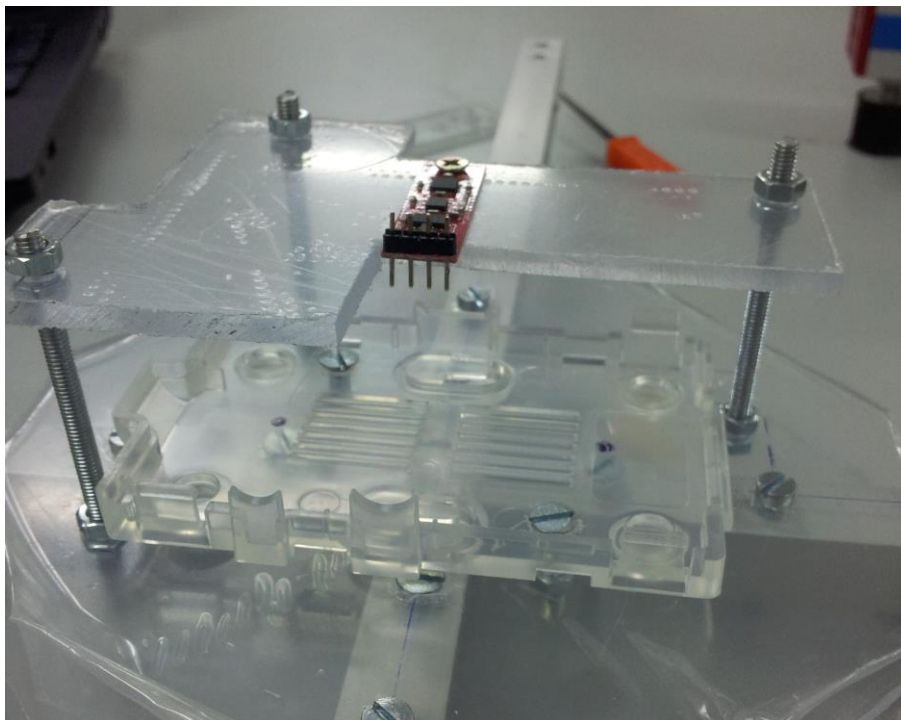


Ilustración 2: Empezando a montar el cuadricóptero

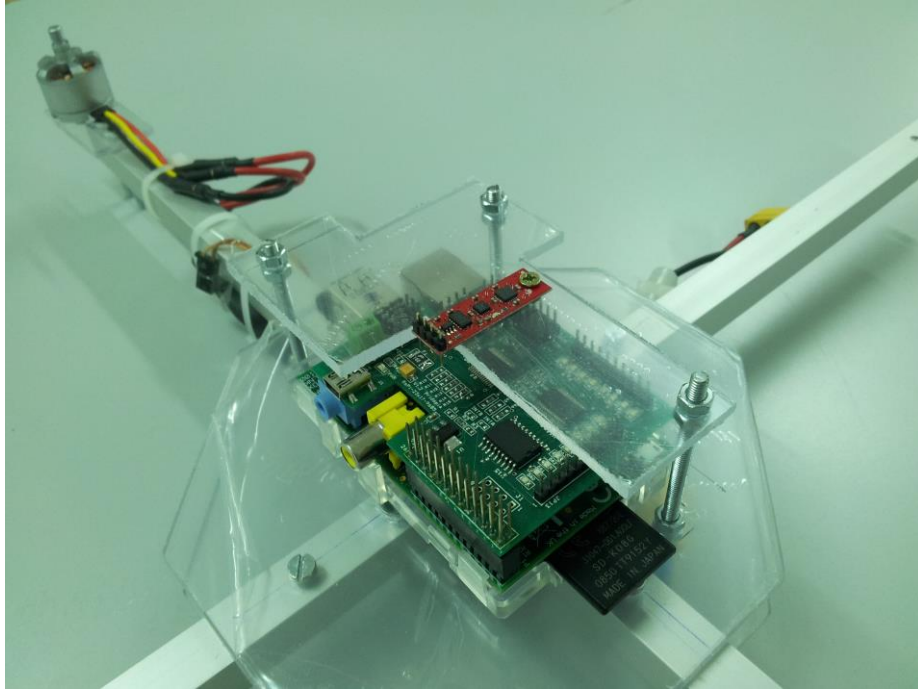


Ilustración 3: Montaje algo más avanzado

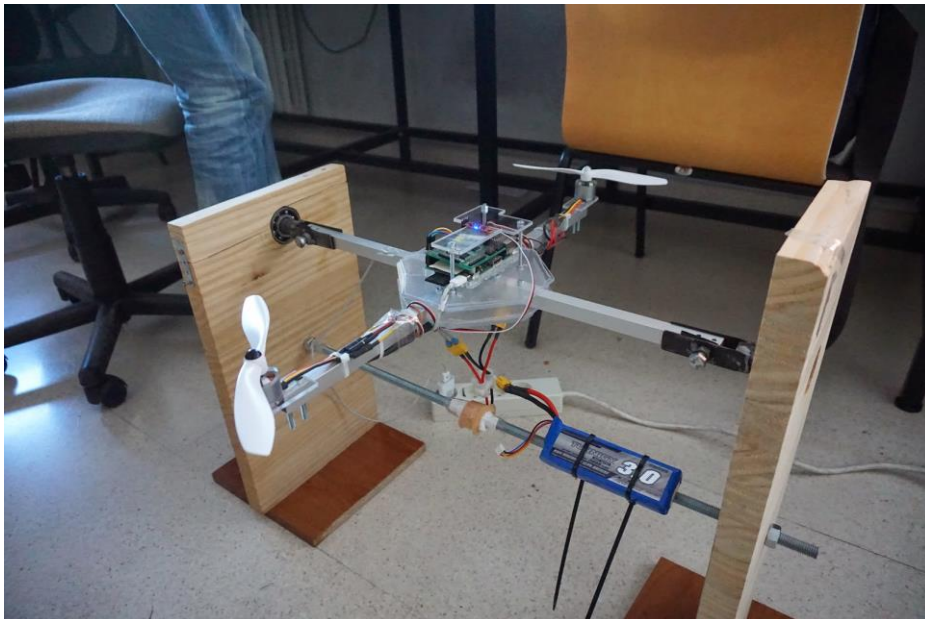


Ilustración 4: Pruebas de control



Ilustración 5: Conjunto motor hélice

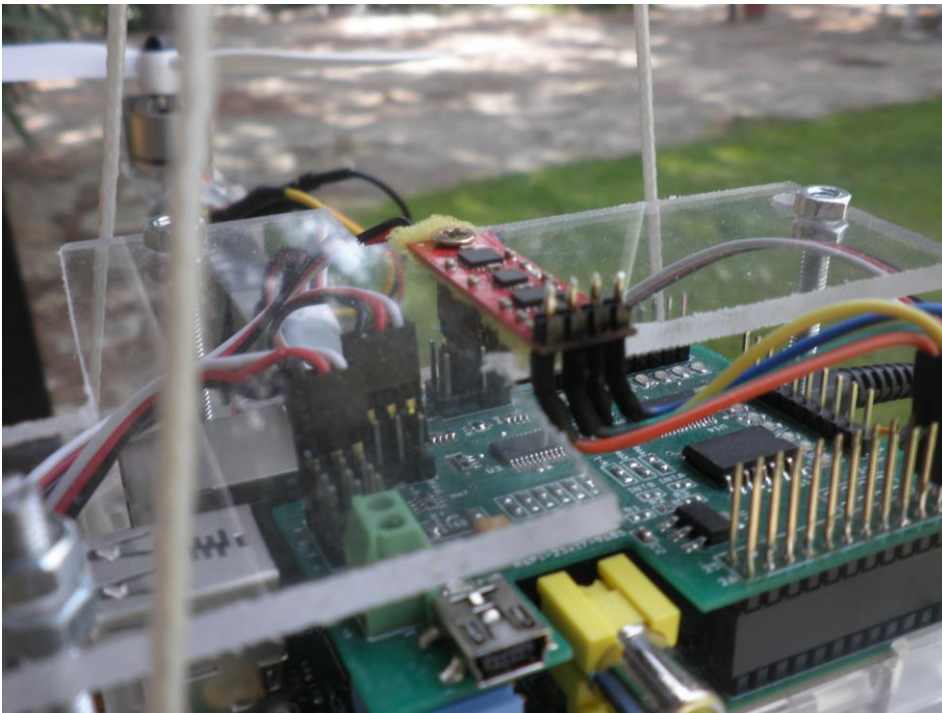


Ilustración 6: Raspberry Pi y sensores



Ilustración 7: Prototipo terminado



Ilustración 8: Pruebas con el prototipo

Videos en <http://cosmecontrol.blogspot.com.es/>

Anexo B: Datasheet de los sensores y el microcontrolador

Adjunto en el CD.

Anexo C: Script de Matlab para la calibración de los motores

Adjunto en el CD.

Anexo D: Código de la aplicación

Adjunto en el CD.