



Proyecto Final de Carrera
INGENIERÍA INDUSTRIAL

Control de un Quadcopter

Memoria

Autor: Gabriel de la Cal Mendoza
Director: Manel Velasco Garcia
Convocatoria: Fecha a presentar

Escola Tècnica Superior d'Enginyeria Industrial de Barcelona



Resumen

En el presente proyecto se ha tenido como meta el control de un Quadcopter cuyas piezas se han adquirido por partes. Para tal propósito se ha linealizado el modelo no lineal obtenido y controlado mediante un Regulador Lineal Cuadrático (LQR). El contenido de este trabajo está organizado en 8 secciones y 4 anexos.

En la sección 1 se presentan los motivos por el que la elección del tema de este PFC queda justificada. También se mencionan requerimientos previos que se han requerido para no tener insalvables dificultades.

En la sección 2, como Introducción, se hace un estudio del arte, con breves pinceladas del basto campo que hoy en día se encuentra sobre este tema. Además se describen los objetivos de este proyecto. Éstos no son más que el resolver todo problema que haya surgido durante esta aventura.

Se obtiene el modelo a partir de los parámetros característicos del sistema físico en la sección 3. En linealizar el sistema en el punto de equilibrio se llega a la conclusión de que las variables que lo forman se desacoplan y se comportan como un conjunto de dobles integradores. Ésto permite que la ley de control sea sencilla y justifica el que se hayan usado controladores basados PID tan asiduamente. También se presenta el modelo implementado en Simulink, con el que se probará el control obtenido posteriormente para verificar su eficiencia.

La sección anterior permite que en la 4 se obtengan las acciones de control $u = -k \cdot X$ por medio de un LQR (Linear Quadratic Regulator). Se ataca al sistema con la ley obtenida juzgando los resultados.

En la sección 5 se explica cómo y qué se ha implementado respecto al control. Se comenta el código C que se está continuamente ejecutando.

En la sección 6 se describe tanto los componentes que se utilizan como el proceso de montaje paso a paso. Se justifica la elección de cada elemento tomando como criterio tanto las características de otros elementos como el dimensionado mismo del Quadcopter.

En las secciones 7, 8 y 9 se tiene un análisis económico, un estudio de impacto ambiental y las conclusiones respectivamente. En el primero se desglosan los costes materiales y se valoran las horas invertidas, considerando que el trabajo lo realiza un estudiante. En el estudio del impacto ambiental se describen los efectos en el medio ambiente de los componentes que se utilizan, tanto en la producción como en el reciclado. En la última sección se relata con qué se queda uno en limpio tras acabar esta empresa y cómo se espera continuar para futuros desarrollos.

En el Anexo 1 explica mediante código Matlab de dónde se ha extraído el vector de fuerzas del modelo. De esta manera se justifica la linealización en ser toda otra opción, en estas circunstancias, inviable.

En el Anexo 2 se detalla el proceso de grabado y posteriores modificaciones realizadas en el sistema operativo utilizado (Raspbian), para condicionar la Raspberry Pi tal y como se ha hecho para el Quadcopter.

En el Anexo 3 se describe qué es un filtro de Kalman, por qué es necesario y cómo se ha implementado.

En el Anexo 4 se describe cómo se han caracterizado los motores y cómo se han obtenido sus curvas características de fuerza y consumo.

Índice

Resum	1
1. Prefacio	5
1.1. Motivación	5
1.2. Requerimientos previos	5
2. Introducción	6
2.1. Estudio del arte	6
2.2. Objetivos del proyecto	7
3. Definición del modelo	8
3.1. Definición de las variables	8
3.2. Obtención del modelo	9
3.3. Representación del modelo con Simulink	14
4. Diseño del controlador	17
4.1. Regulador Quadrático Lineal	17
4.2. Obtención de la ley de control	17
4.3. Aplicación en el modelo	18
5. Implementación del control	19
6. Construcción del Quadcopter	20
6.1. Descripción de los componentes	20
6.2. Montaje	25
7. Análisis económico	26
8. Estudio del impacto ambiental	27
9. Conclusiones	28

Índice de figuras

1.	Quadcopter de ejemplo con cámara	6
2.	Ejemplo de Quadcopter con 6 motores	6
3.	Marcos de referencia en el quadcopter	8
4.	Modelo en Simulink de la dinámica del Quadcopter	14
5.	Raspberry Pi a utilitzar	20
6.	Conectors de la Raspberry Pi	20
7.	IMU MPU-6050	21
8.	Emissora i Receptor	21
9.	Esquema de un Variador como inversor trifásico	23
10.	Tabla de Set up de un ESC	23
11.	Piezas que componen la estructura del Quadcopter	24
12.	Par de hélices 8x4 simétricas	24
13.	Representación de la Longitud y Pitch de una Hélice	24
14.	Conexión por I2C entre MPU-6050 y RPi	33
15.	Detección del sensor MPU-6050	34
16.	Bancada para el estudio del motor	36
17.	Montaje para el estudio del motor	36
18.	Relación entre la fuerza (en gramos) y el PWM	38

1. Prefacio

1.1. Motivación

La principal motivación de este proyecto es la de aplicar por uno mismo los conocimientos básicos adquiridos en la carrera, y más en particular en el área del control en haber cursado la intensificación de automática.

En plantear un tema para el proyecto rápidamente surgió la idea de realizarlo sobre el control de un sistema mecánico, y más en particular sobre uno que estuviera actualmente emergiendo tanto en mercados como en el campo de la investigación.

De entre las diferentes alternativas, un quadcopter es la más atractiva para el proyecto tanto por su simplicidad constructiva como por su no excesivo coste. Existen actualmente en el mercado infinidad de proveedores para los componentes necesarios para construir un quadcopter, con un gran abanico de opciones de entre las que escoger cada elemento como motores, baterías, electrónica, etc.

Las posibles aplicaciones son numerosas, tanto que aún no se han ni tan solo explotado todas las posibles. Como ejemplo: vigilancia de superficies abiertas, transporte de pequeños paquetes, herramienta de ocio, entre otras.

Entender el funcionamiento y familiarizarse con el mecanismo es una inquietud que se ha mantenido insatisfecha durante largo tiempo. Con este pretexto se pretende acabar de justificar la elección del tema de ésta pequeña gran empresa.

1.2. Requerimientos previos

Es necesario tener ciertos conocimientos mínimos en automática para poder controlar el quadcopter, así como la inquietud de aprender lo que sea necesario para cumplir, en la medida de lo posible, los objetivos iniciales del presente proyecto.

Nociones de mecánica son necesarias para elaborar e interpretar el sistema sobre el que se trabaje, así como sus resultados. Conocimientos del lenguaje *C* son imprescindibles ya que el código del programa que realiza los cálculos está escrito en este mismo lenguaje.

Para adaptar los elementos que componen el aparato y afianzarlos a la estructura (Frame) del Quadcopter ha sido de gran ayuda contar con el servicio de una impresora 3D, creada por uno mismo. Finalmente, estar familiarizado con el argot y elementos de un Quadcopter, así como motores, emisoras de radio, entre otros es de ayuda.

Superar las dificultades que han surgido, teniendo en cuenta que el fin no justifica los medios, es la praxis que se ha aplicado a la hora de realizar éste trabajo. Finalmente, recordar reminiscencias de la carrera es siempre útil e interesante, así que conceptos adquiridos pero no utilizados se esperan aplicar.

2. Introducción

Un Quadcopter es un vehículo volador no tripulado (*Unmanned Aerial Vehicle* o *UAV*) que se caracteriza por tener cuatro rotores a modo de actuadores en vez de dos como en el caso de los helicópteros. Este tipo de autogiro intenta obtener una flotabilidad estable y vuelo preciso balanceando las fuerzas producidas por los cuatro motores y surge de la necesidad de tener un aparato de vuelo de dimensiones reducidas y gran estabilidad, con prestaciones superiores a las de un helicóptero convencional.



Figura 1: Quadcopter de ejemplo con cámara

Una de las ventajas que se obtiene con este cambio es la mayor capacidad de carga ya que es tienen 4 motores para soportar el peso. La estabilidad del vehículo mejora en permitir aterrizajes y despegues verticales con una mayor maniobrabilidad. También puede trabajar en áreas de difícil acceso o más agresivas, como con lluvia y viento.

Esquemáticamente se puede representar como una estructura en *X* con su centroide coincidiendo con el centro de masas i cuatro actuadores a las puntas de cada brazo, todos ellos apuntando en la misma dirección y sentido, pero con giros de aspa en sentido contrario, pero igual en lados opuestos.

2.1. Estudio del arte

En la actualidad el campo referente a estos aparatos se ha diversificado tanto que se pueden encontrar muchas variedades y tipos de Quadcopters. Se ha hecho mención del modelo de cuatro motores, pero bien pueden encontrarse de tres hasta 8 motores, sino más en casos más concretos.



Figura 2: Ejemplo de Quadcopter con 6 motores

Con el objetivo de desarrollar un *UAV* que realice tareas de forma totalmente autónoma, se han diseñado sensores para estimar los estados del vehículo de manera más rápida y eficiente, así como diferentes estrategias para controlar la estabilidad y orientación del mismo. Se ha utilizado visión por computador para realizar aterrizajes, detección de objetivos y navegación autónoma. También pueden combinarse diferentes técnicas con el uso del GPS para asegurar

una buena orientación del aparato.

Para controlar esta clase de aparatos se utilizan diferentes leyes de control, así como PID, LQR, Redes Neuronales, Machine Learning, entre otros. La más utilizada es sin duda un PID por su fácil implantación y poca carga computacional.

2.2. Objetivos del proyecto

La meta que se persigue es la de hacer planear un Quadcopter en direcciones horizontales. A la vez, éste es un excelente subterfugio para cumplir con otros objetivos relacionados con este fin como el control de motores, el inteligente uso de la electrónica y correcto diseño mecánico del aparato. Además, se espera haber ampliado la visión que uno tiene en cada campo.

De entre todas las disciplinas involucradas, se espera acabar familiarizado con las facetas relacionadas con la presente empresa para clarificar la relación entre los elementos que componen un Quadcopter. A saber, se espera poder acabar entendiendo la electrónica, mecánica y conocimientos de control que reinan en el sistema.

En el caso de la electrónica se espera entender y poder implementar sencillos protocolos de comunicación, en este caso *I2C* entre los sensores y la Raspberry Pi. Ésta misma es el centro neurálgico del Quadcopter, así que es esencial saber utilizarla como se requiere para que pueda recibir consignas, estados y enviar la correspondiente acción de control a los motores para que se llegue al estado final deseado.

Por parte de la mecánica, es primordial tener conocimientos sólidos de conceptos como el Teorema de la Cantidad de Movimiento y el de la conservación del Momento de Inercia, así como poder calcular el Lagrangiano (\mathcal{L}) para generar un modelo verosímil y que represente fidedignamente la realidad para que se desarrolle de manera efectiva el control.

Finalmente, y como más serio objetivo de toda esta aventura, es el de haber afianzado lo que se creía entendido en relación al control. Asimilar otros métodos de control aplicables a sistemas multilineales no contemplados en la carrera es un excitante reto que se espera haber superado con éxito.

Entonces, se entiende este trabajo como la causa de la motivación a cumplir con un objetivo multidisciplinar como puede serlo cualquiera en el campo de la robótica, en el que se debe tener constancia de muchos aspectos relevantes, como el de la economía. Poder tener idea de lo que cuesta un robot es de gran importancia si uno se embarca en un proyecto de envergadura, ya que se quiere que los pronósticos sobre el presupuesto cuadren con la futura realidad.

3. Definición del modelo

3.1. Definición de las variables

Para caracterizar la planta con la que es trabajarán, es necesario obtener un modelo del Quadcopter. Las constantes propias del modelo se dejarán en forma de parámetros a calibrar una vez se tenga el objeto físico. De esta manera el modelo será general para todo quadcopter que comparta la misma familia de parámetros. Es necesario considerar dos marcos de referencia: el inercial formado por los ejes x, y, z y el del cuerpo (Body) formado por los ejes x_B, y_B, z_B . El primero tiene la perspectiva de el observador en tierra, estático, mientras que el segundo es solidario a la estructura. Según la orientación de los ejes del cos con esta referencia se pueden dar los siguientes dos casos:

- **Cross type:** Los ejes de coordenadas coinciden con los brazos de la estructura ya que se tienen los actuadores a las puntas de cada brazo.
- **X-type:** Los ejes y la estructura forman 45. Se tienen entonces dos motores en la parte delantera y dos en la trasera.

Por ser más usual la primera opción, se decide utilizar la configuración *Cross type* tal y como se tiene en la figura 3.

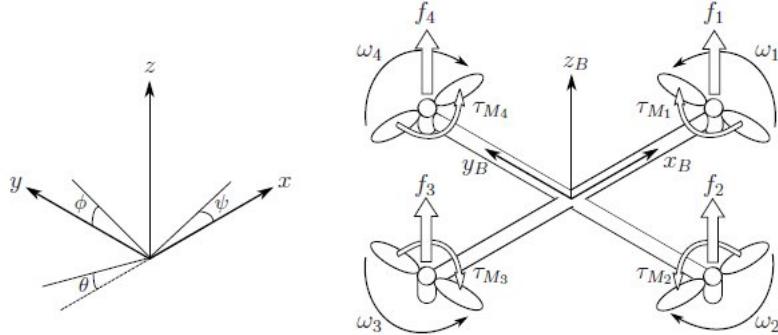


Figura 3: Marcos de referencia en el quadcopter

Se supone que el objeto es un rotor esférico, y por tanto su tensor de inercia es diagonal:

$$\mathbf{I} = \begin{bmatrix} I_{xx} & 0 & 0 \\ 0 & I_{yy} & 0 \\ 0 & 0 & I_{zz} \end{bmatrix} \quad (1)$$

Se define la posición lineal absoluta con las coordenadas x, y, z con vector ξ e igualmente para la posición angular a partir de η según:

$$\xi = \begin{bmatrix} x \\ y \\ z \end{bmatrix}, \quad \eta = \begin{bmatrix} \phi \\ \theta \\ \psi \end{bmatrix}, \quad \mathbf{q} = \begin{bmatrix} \xi \\ \eta \end{bmatrix} \quad (2)$$

donde ϕ es el ángulo de cabeceo (Pitch), θ es el de balanceo (Roll) y ψ el de guiñada (Yaw). Para la orientación angular entre los dos marcos se tiene un sistema de referencia con ángulos Tait-Bryan, donde la matriz de transformación es:

$$\mathbf{R} = \begin{bmatrix} C_\psi C_\theta & C_\psi S_\theta S_\phi - S_\psi C_\phi & C_\psi S_\theta C_\phi + S_\psi S_\phi \\ S_\psi C_\theta & S_\psi S_\theta S_\phi + C_\psi C_\phi & S_\psi S_\theta C_\phi - C_\psi S_\phi \\ -S_\theta & C_\theta C_\phi & C_\theta S_\phi \end{bmatrix} \quad (3)$$

con $C_\phi = \cos(\phi)$ i $S_\phi = \sin(\phi)$.

Las velocidades lineales en el marco de referencia del cuerpo (Body Frame) se representen con el vector v_B y las velocidades angulares con γ según:

$$\mathbf{v}_B = \begin{bmatrix} v_{x,b} \\ v_{y,b} \\ v_{z,b} \end{bmatrix} \quad \boldsymbol{\gamma} = \begin{bmatrix} p \\ n \\ r \end{bmatrix} \quad (4)$$

En cambio, las velocidades en el marco de referencia inercial (Inertial Frame) se representan por $\dot{\eta}$ para las velocidades lineales y con $\dot{\xi}$ para las angulares:

$$\dot{\xi} = \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \end{bmatrix} \quad \dot{\eta} = \begin{bmatrix} \dot{\psi} \\ \dot{\theta} \\ \dot{\phi} \end{bmatrix} \quad (5)$$

Ya que la derivada de los ángulos ϕ , θ y ψ no es el vector de velocidades angulares es necesario tener un cambio de base para relacionar el marco de referencia inercial con el del cuerpo con la matriz W_η :

$$\mathbf{W}_\eta = \begin{bmatrix} 1 & 0 & -S_\theta \\ 0 & C_\phi & C_\theta S_\phi \\ 0 & -S_\phi & C_\theta C_\phi \end{bmatrix} \quad amb \quad \boldsymbol{\gamma} = [\mathbf{W}_\eta] \dot{\eta} \quad (6)$$

Las fuerzas de sustentación y velocidades angulares de los cuatro actuadores son f_1, f_2, f_3, f_4 i w_1, w_2, w_3, w_4 respectivamente. Siguiendo la orientación de la figura 3, con el objetivo de poder anular los momentos producidos en el eje z_B (en el marco del cuerpo B) el sentido de giro de los actuadores 4 y 2 son en el de las agujas del reloj (clockwise) y el de los 1 y 3 en sentido contrario (counterclockwise).

Interesa conocer qué fuerza y momento aportará cada motor dada una velocidad angular conocida. Se supone poder controlar la fuerza y momento ejercida por cada motor y cómo aproximación inicial se considera que están relacionados con la velocidad angular de la forma:

$$\begin{aligned} f_i &= kw_i^2 \\ \tau_{M_i} &= bw_i^2 \end{aligned} \quad (7)$$

Por tanto el empuje total \mathbf{T}_B proporcionado en la dirección z_B y los momentos generados $\boldsymbol{\tau}_B$ por los motores son:

$$\mathbf{T}_B = k \left(\sum_{i=1}^4 w_i^2 \right) e_{z_B} = \begin{bmatrix} 0 \\ 0 \\ \sum_{i=1}^4 f_i \end{bmatrix} \quad \boldsymbol{\tau}_B = \begin{bmatrix} \tau_\phi \\ \tau_\theta \\ \tau_\psi \end{bmatrix} = \begin{bmatrix} l(f_4 - f_2) \\ l(f_3 - f_1) \\ \sum_{i=1}^4 \tau_{M_i} \end{bmatrix} \quad (8)$$

3.2. Obtención del modelo

Las ecuaciones que gobiernan el sistema se obtienen con el método de Euler-Lagrange, por lo que se empieza obteniendo el Lagrangiano del sistema:

$$\mathcal{L} = E_{cinetica} - E_{potencial} = (E_{translacion} + E_{rotacion}) - E_{potencial} \quad (9)$$

Substituyendo cada componente por su expresión:

$$\mathcal{L}(\mathbf{q}, \dot{\mathbf{q}}) = \frac{m}{2} \dot{\xi}^T \dot{\xi} + \frac{1}{2} \boldsymbol{\gamma}^T \mathbf{I} \boldsymbol{\gamma} - mgz \quad (10)$$

Es encuentra el vector de fuerzas y momentos:

$$\mathbf{F} = \begin{bmatrix} \mathbf{f} \\ \boldsymbol{\tau}_B \end{bmatrix} = \frac{d}{dt} \left(\frac{\partial \mathcal{L}}{\partial \dot{\mathbf{q}}} \right) - \frac{\partial \mathcal{L}}{\partial \mathbf{q}} \quad (11)$$

$$\text{amb} \quad \mathbf{q} = [x \ y \ z \ \phi \ \theta \ \psi]^T \quad i \quad \dot{\mathbf{q}} = [\dot{x} \ \dot{y} \ \dot{z} \ \dot{\phi} \ \dot{\theta} \ \dot{\psi}]^T.$$

En calcular F es necesario hacer el cambio de variables de γ a $\dot{\eta}$ con el cambio de base $\dot{\eta} = [W_\eta]^{-1} \gamma$ para poder derivar el Lagrangiano respecto \dot{q} , que son las variables propias del marco de referencia inercial:

$$\frac{1}{2} \boldsymbol{\gamma}^T \mathbf{I} \boldsymbol{\gamma} = \frac{1}{2} (\mathbf{W}_\eta \dot{\boldsymbol{\eta}})^T \mathbf{I} (\mathbf{W}_\eta \dot{\boldsymbol{\eta}}) = \frac{1}{2} \dot{\boldsymbol{\eta}}^T (\mathbf{W}_\eta^T \mathbf{I} \mathbf{W}_\eta) \dot{\boldsymbol{\eta}} = \frac{1}{2} \dot{\boldsymbol{\eta}}^T \mathbf{J} \dot{\boldsymbol{\eta}} \quad (12)$$

donde la matriz J queda como

$$\mathbf{J} = \mathbf{W}_\eta^T \mathbf{I} \mathbf{W}_\eta = \begin{bmatrix} I_{xx} & 0 & -I_{xx} S_\theta \\ 0 & I_{yy} C_\phi^2 + I_{zz} S_\phi^2 & (I_{yy} - I_{zz}) C_\phi S_\phi C_\theta \\ -I_{xx} S_\theta & (I_{yy} - I_{zz}) C_\phi S_\phi C_\theta & I_{xx} S_\theta^2 + I_{yy} S_\phi^2 C_\theta^2 + I_{zz} C_\phi^2 C_\theta^2 \end{bmatrix} \quad (13)$$

y por tanto el Lagrangiano queda:

$$\mathcal{L}(\mathbf{q}, \dot{\mathbf{q}}) = \frac{m}{2} \dot{\boldsymbol{\xi}}^T \dot{\boldsymbol{\xi}} + \frac{1}{2} \dot{\boldsymbol{\eta}}^T \mathbf{J} \dot{\boldsymbol{\eta}} - mgz \quad (14)$$

Los componentes lineales y angulares no dependen unos de los otros y por tanto se pueden estudiar por separado obteniendo dos ecuaciones: una para las fuerzas lineales y otro para los momentos. Esto quiere decir que la fuerza que ejercen los actuadores no depende de las velocidades angulares que se tengan ni tampoco se tendrán aceleraciones angulares diferentes según la altura a la que se encuentre el quadcopter: el objeto girará de la misma manera sea cual sea la posición en la que se encuentre en el espacio.

Entonces, calculando la derivada parcial respecto \dot{q} se obtiene que

$$\mathbf{F} = \frac{d}{dt} \left(\frac{m}{2} (1 \cdot \dot{\boldsymbol{\xi}} + \dot{\boldsymbol{\xi}} \cdot 1) + \frac{1}{2} \frac{\partial}{\partial \dot{\boldsymbol{\eta}}} (\dot{\boldsymbol{\eta}}^T \mathbf{J} \dot{\boldsymbol{\eta}}) \right) - \frac{\partial \mathcal{L}}{\partial \mathbf{q}} \quad (15)$$

Como que \mathbf{J} es una matriz simétrica, se puede decir que $\frac{\partial}{\partial \dot{\boldsymbol{\eta}}} (\dot{\boldsymbol{\eta}}^T \mathbf{J} \dot{\boldsymbol{\eta}}) = 2 \frac{\partial}{\partial \dot{\boldsymbol{\eta}}} (\dot{\boldsymbol{\eta}}^T \mathbf{J}) \dot{\boldsymbol{\eta}}$.

Demostración: Para probar ésto se verá para el caso $\frac{\partial}{\partial \mathbf{x}} (\mathbf{x}^T \mathbf{A} \mathbf{x}) = 2 \frac{\partial}{\partial \mathbf{x}} (\mathbf{x}^T \mathbf{A}) \mathbf{x}$ con

$$\mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} \quad \mathbf{A} = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix} \quad (16)$$

Llavors

$$\frac{\partial}{\partial \mathbf{x}} (\mathbf{x}^T \mathbf{A} \mathbf{x}) = \frac{\partial}{\partial \mathbf{x}} \left([\mathbf{x}_1 \mathbf{x}_2 \mathbf{x}_3] \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} \right) = \quad (17)$$

$$\frac{\partial}{\partial \mathbf{x}} (x_1^2 a_{11} + x_1 x_2 a_{21} + x_1 x_3 a_{31} + x_1 x_2 a_{12} + x_2^2 a_{22} + x_2 x_3 a_{32} + x_1 x_3 a_{13} + x_2 x_3 a_{23} + x_3^2 a_{33}) = \quad (18)$$

Com que \mathbf{A} es simétrica $a_{12} = a_{21}$, $a_{13} = a_{31}$ i $a_{23} = a_{32}$, y en hacer la derivada direccional resulta

$$\frac{\partial}{\partial \mathbf{x}} (\mathbf{x}^T \mathbf{A} \mathbf{x}) = \begin{bmatrix} 2x_1 a_{11} + x_2 a_{21} + x_3 a_{31} + x_2 a_{12} + x_3 a_{13} \\ x_1 a_{21} + x_1 a_{12} + 2x_2 a_{22} + x_3 a_{32} + x_3 a_{23} \\ x_1 a_{31} + x_2 a_{32} + x_1 a_{31} + x_2 a_{23} + 2x_3 a_{23} \end{bmatrix} = 2 \cdot \begin{bmatrix} x_1 a_{11} + x_2 a_{12} + x_3 a_{13} \\ x_1 a_{12} + x_2 a_{22} + x_3 a_{23} \\ x_1 a_{13} + x_2 a_{23} + x_3 a_{23} \end{bmatrix} \quad (19)$$

Y en evaluar el otro costado de la igualdad se tiene el mismo resultado

$$2 \frac{\partial}{\partial \mathbf{x}} (\mathbf{x}^T \mathbf{A}) \mathbf{x} = 2 \frac{\partial}{\partial \mathbf{x}} \left(\begin{bmatrix} x_1 a_{11} + x_2 a_{12} + x_3 a_{13} \\ x_1 a_{21} + x_2 a_{22} + x_3 a_{23} \\ x_1 a_{31} + x_2 a_{32} + x_3 a_{33} \end{bmatrix}^T \right) \mathbf{x} = \quad (20)$$

$$= 2 \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix} \cdot \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = 2 \cdot \begin{bmatrix} x_1 a_{11} + x_2 a_{12} + x_3 a_{13} \\ x_1 a_{12} + x_2 a_{22} + x_3 a_{23} \\ x_1 a_{13} + x_2 a_{23} + x_3 a_{23} \end{bmatrix} \quad (21)$$

Como que $2\frac{\partial}{\partial \dot{\eta}}(\dot{\eta}^T J)\dot{\eta} = 2J\ddot{\eta}$ se tiene, aplicando la regla de la cadena en el producto $J\ddot{\eta}$:

$$\mathbf{F} = \frac{d}{dt}(m\dot{\xi} + \mathbf{J}\dot{\eta}) - \frac{\partial \mathcal{L}}{\partial \mathbf{q}} = m\ddot{\xi} + \mathbf{J}\ddot{\eta} + \dot{\mathbf{J}}\dot{\eta} - \left(\frac{1}{2} 2 \frac{\partial}{\partial \dot{\eta}}(\dot{\eta}^T \mathbf{J})\dot{\eta} - mg \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \right) \quad (22)$$

Para llegar a este resultado se ha aplicado la derivada direccional a mgz :

$$D_q(mgz) = D_{\xi}(mgz) = mg \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \quad (23)$$

Separando las componentes lineales y angulares en dos ecuaciones:

$$f = m\ddot{\xi} + mg \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} = \mathbf{R}\mathbf{T}_B \quad (24)$$

$$\tau = \mathbf{J}\ddot{\eta} + \underbrace{\left(\dot{\mathbf{J}} - \frac{\partial}{\partial \dot{\eta}}(\dot{\eta}^T \mathbf{J}) \right) \dot{\eta}}_{\mathbf{C}(\eta, \dot{\eta})} = \mathbf{J}\ddot{\eta} + \mathbf{C}(\eta, \dot{\eta}) \cdot \dot{\eta} \quad (25)$$

Donde $\mathbf{C}(\eta, \dot{\eta})$ es la matriz de Coriolis. Para obtener el sistema de ecuaciones del modelo se deben aislar las aceleraciones y se obtiene:

$$\begin{cases} \ddot{\xi} &= \frac{1}{m}\mathbf{R}\mathbf{T}_B - g \cdot [0 \ 0 \ 1]^T \\ \ddot{\eta} &= \mathbf{J}^{-1}(\tau - \mathbf{C}(\eta, \dot{\eta}) \cdot \dot{\eta}) \end{cases} \quad (26)$$

Reescribiendo estas ecuaciones de la forma $\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, \mathbf{u})$, donde \mathbf{u} es el conjunto de fuerzas ejercidas por los motores, se tiene

$$\frac{\partial}{\partial t} \begin{bmatrix} \xi \\ \dot{\xi} \\ \eta \\ \dot{\eta} \end{bmatrix} = \begin{bmatrix} \ddot{\xi} \\ \frac{1}{m}\mathbf{R}\mathbf{T}_B - g \cdot [0 \ 0 \ 1]^T \\ \ddot{\eta} \\ \mathbf{J}^{-1}(\tau - \mathbf{C}(\eta, \dot{\eta}) \cdot \dot{\eta}) \end{bmatrix} \quad (27)$$

Para realizar el control será útil representar el sistema 26 en forma de espacio de estados, y por tanto el vector de estados será de la forma $\mathbf{X} = [x \ \dot{x} \ y \ \dot{y} \ z \ \dot{z} \ \phi \ \dot{\phi} \ \theta \ \dot{\theta} \ \psi \ \dot{\psi}]$. El punto de equilibrio de este sistema es aquél que hace que las componentes angulares del vector de estados no varíe y no haya desplazamientos lineales. Éste caso se puede dar para toda posición ξ dada. Ésto se tiene si todos los motores hacen exactamente la misma fuerza y entre todos cuatro la misma al peso del Quadcopter, además de tener un momento de inercia nulo y evitar que el ángulo ψ (yaw) varíe. Se trata de un punto de equilibrio forzado. El punto de equilibrio será entonces:

$$\begin{cases} \mathbf{X}_0 = [x \ 0 \ y \ 0 \ z \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0]^T \\ \mathbf{U}_0 = \begin{bmatrix} f_1 \\ f_2 \\ f_3 \\ f_4 \end{bmatrix} = \frac{m \cdot g}{4} \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{bmatrix} \end{cases} \quad (28)$$

El modelo se expresa en forma lineal como

$$\begin{aligned} \dot{\mathbf{X}} &= [\mathbf{A}] \cdot \mathbf{X} + \mathbf{K} + [\mathbf{B}] \cdot \mathbf{U} \\ \mathbf{Y} &= [\mathbf{C}] \cdot \mathbf{X} \end{aligned} \quad (29)$$

donde \mathbf{K} es el termino que incluye a la constante de la gravedad, separada de la función $\mathbf{f}(\mathbf{x}, \mathbf{u})$. Se calculan las matrices \mathbf{A} y \mathbf{B} con los respectivos jacobianos en el punto de equilibrio:

$$\mathbf{A} = \frac{\partial \mathbf{f}(\mathbf{x}, \mathbf{u})}{\partial \mathbf{x}} \Big|_{\mathbf{x}_0, \mathbf{U}_0} \quad \mathbf{B} = \frac{\partial \mathbf{f}(\mathbf{x}, \mathbf{u})}{\partial \mathbf{u}} \Big|_{\mathbf{x}_0, \mathbf{U}_0} \quad (30)$$

En considerar que las posiciones angulares de ϕ y θ no diferirán significativamente respecto de zero y que el ángulo ψ no es relevante, se trabaja con una matriz de rotación R igual a la identidad

$$\mathbf{R} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (31)$$

Se ha mantenido el término independiente de la gravedad a fuera de la función $\mathbf{f}(\mathbf{x}, \mathbf{u})$ para no eliminarlo en la derivada del jacobiano. Las matrices \mathbf{A} y \mathbf{B} quedan como

$$\mathbf{A} = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \quad \mathbf{B} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 1/m & 1/m & 1/m & 1/m \\ 0 & 0 & 0 & 0 \\ 0 & -l/I_{xx} & 0 & l/I_{xx} \\ 0 & 0 & 0 & 0 \\ -l/I_{yy} & 0 & l/I_{yy} & 0 \\ 0 & 0 & 0 & 0 \\ b/I_{zz} & -b/I_{zz} & b/I_{zz} & -b/I_{zz} \end{bmatrix} \quad (32)$$

y las matrices \mathbf{K} i \mathbf{C} quedan como

$$\mathbf{K} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ -9,81 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \quad \mathbf{C} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix} \quad (33)$$

Éstos son los resultados sacados del procedimiento explicado en el *Anexo 1*, en donde se detalla el procedimiento.

Como se desea implantar un control sobre el equilibrio del Quadcopter en vez de la posición, se tendrán las variables pertinentes a la orientación, y por tanto el vector de estados estará formado por $[\dot{\phi} \ddot{\phi} \dot{\theta} \ddot{\theta}]$.

Queda entonces

$$\begin{bmatrix} \dot{\phi} \\ \ddot{\phi} \\ \dot{\theta} \\ \ddot{\theta} \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \dot{\phi} \\ \ddot{\phi} \\ \dot{\theta} \\ \ddot{\theta} \end{bmatrix} + \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & -l/I_{xx} & 0 & l/I_{xx} \\ 0 & 0 & 0 & 0 \\ -l/I_{yy} & 0 & l/I_{yy} & 0 \end{bmatrix} \begin{bmatrix} f_1 \\ f_2 \\ f_3 \\ f_4 \end{bmatrix} \quad (34)$$

$$\mathbf{Y} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} \dot{\phi} \\ \dot{\psi} \\ \theta \\ \dot{\theta} \end{bmatrix}$$

donde el vector \mathbf{K} se ha obviado porque las componentes que se hubieran considerado de éste son todas nulas.

El eliminar estas componentes se hace porque no hay una referencia zero para el ángulo ψ , mientras que en el caso de los ángulos ϕ y θ la referencia siempre será la horizontal. No supone un infranqueable problema a la hora de hacer mover el Quadcopter, pues éste ya se puede desplazar en todo su plano horizontal, sin necesidad de girar en la dirección del *yaw*.

Para abordar el control de este sistema reducido se evalúa su controlabilidad y observabilidad. Se espera poder llevar el vector de estados de una posición a otra mediante una entrada adecuada, en un tiempo finito. La matriz de controlabilidad $\mathbf{W}_c = [B \ AB \ A^2B \ \dots]$ es de la forma:

$$\mathbf{W}_c = \left[\begin{array}{cccc|cccc|c} 0 & 0 & 0 & 0 & 0 & -l/I_{xx} & 0 & l/I_{xx} & \dots \\ 0 & -l/I_{xx} & 0 & l/I_{xx} & 0 & 0 & 0 & 0 & \dots \\ 0 & 0 & 0 & 0 & -l/I_{yy} & 0 & l/I_{yy} & 0 & \dots \\ -l/I_{yy} & 0 & l/I_{yy} & 0 & 0 & 0 & 0 & 0 & \dots \end{array} \right] \quad (35)$$

Con las 8 primeras columnas es suficiente para ver que el rango de la matriz es máximo ya que el resto de componentes de la matriz son nulas. Por lo tanto, no hay subsistema inaccesible des de la entrada y es entonces controlable:

$$\text{rango}(\mathbf{W}_c) = 4 \ (\text{max.}) \rightarrow \text{Controlable} \quad (36)$$

Para evaluar la observabilidad se calcula la matriz de observabilidad \mathbf{W}_o y como en el caso anterior, se evaluará su rango:

$$\mathbf{W}_o = \left[\begin{array}{c} C \\ CA \\ CA^2 \\ \dots \end{array} \right] = \left[\begin{array}{cccc} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ \dots & & & \end{array} \right] \quad (37)$$

Otra vez no es necesario evaluar todos los términos de la matriz, ya que con las primeras 4 filas se ve que el rango vuelve a es máximo:

$$\text{rango}(\mathbf{W}_o) = 4 \ (\text{max.}) \rightarrow \text{Observable} \quad (38)$$

Ésto arroja lo que ya era sabido del sistema por su misma naturaleza física, y es que de un Quadcopter se pueden observar sus estados y que se puede gobernar (controlar) mediante los actuadores provistos.

3.3. Representación del modelo con Simulink

Obtenidas las ecuaciones del modelo, se quiere poder trabajar con éste para comprobar que la ley de control que se tenga es válida como para intentar controlar el aparato real. Se intenta que el modelo sea lo más preciso posible, por lo que no se tienen en cuenta linealizaciones ni aproximaciones de tipo alguno.

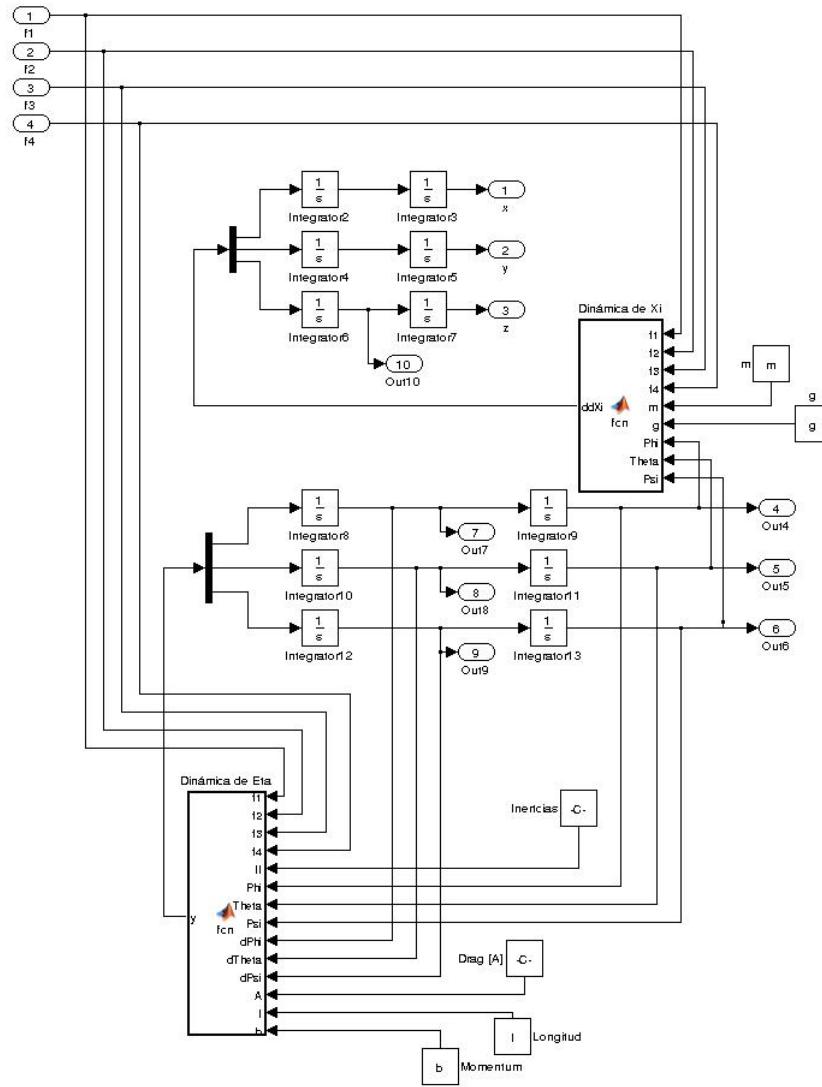


Figura 4: Modelo en Simulink de la dinámica del Quadcopter

Aunque en el modelo se hayan tenido en cuenta los respectivos coeficientes de fricción con el aire, como se ha visto en el apartado anterior no se han tenido en cuenta a la hora de linealizar el sistema ni calcular la ley de control ya que la determinación de estos parámetros no es fácil y de incluirse podría aportar incertidumbre.

Tal y como se explica a la hora de caracterizar los motores, no se pudo encontrar la relación existente que la fuerza y momento ejercido por cada hélice, por lo que en este caso se ha supuesto que varía proporcionalmente con b .

Los dos bloques presentes en el modelo (*Function Matlab*) definen su dinámica. La función *Dinmica de Xi* determina las aceleraciones lineales. El código que se ejecuta en ésta es:

```

1 function ddXi = fcn(f1,f2,f3,f4,m,g,Phi,Theta,Psi)
2 %codegen
3
4 R=[[cos(Psi)*cos(Theta) cos(Psi)*sin(Theta)*sin(Phi)-sin(Psi)*cos(Phi)
5 cos(Psi)*sin(Theta)*cos(Phi)+sin(Psi)*sin(Phi)];
6 [sin(Psi)*cos(Theta) sin(Psi)*sin(Theta)*sin(Phi)+cos(Psi)*cos(Phi)
7 sin(Psi)*sin(Theta)*cos(Phi)-cos(Psi)*sin(Phi)];
8 [-sin(Theta) cos(Theta)*sin(Phi) cos(Theta)*cos(Phi)]];
9
10 T=[0; 0; (f1+f2+f3+f4)/m];
11 ddXi = R*T/m+[0; 0; -g];

```

En ella se calcula la matriz de rotación R en función de la posición angular (ϕ, θ y ψ) y con el vector de fuerzas en la base móvil (del cuerpo *Body*) se calcula en vector de aceleraciones lineales, es decir, $\ddot{\xi} = [\ddot{\phi} \quad \ddot{\theta} \quad \ddot{\psi}]^T$.

En la función *Dinmica de Eta* se calculan las aceleraciones angulares. Su código corresponde a:

```

1 function y = fcn(f1,f2,f3,f4,II,Phi,Theta,Psi,dPhi,dTheta,dPsi,A,l,b)
2 %codegen
3
4 Ixx=II(1);
5 Iyy=II(2);
6 Izz=II(3);
7
8 TauB=[[ l*(f4-f2)];
9 [ l*(f3-f1)];
10 [ b*(f1+f3-f2-f4) ]];
11
12 Weta=[[ 1 0 -sin(Theta)];
13 [ 0 cos(Phi) cos(Theta)*sin(Phi)];
14 [ 0 -sin(Phi) cos(Theta)*cos(Phi) ]];
15
16 J=transpose(Weta)*[[Ixx 0 0];[0 Iyy 0];[0 0 Izz]]*Weta;
17
18 CdE1=(Iyy*dTheta^2*sin(2*Phi))/2 - (Izz*dTheta^2*sin(2*Phi))/2 - Ixx
19 *dPsi*dTheta*cos(Theta) - (Iyy*dPsi^2*sin(2*Phi)*cos(Theta)^2)/2
+ (Izz*dPsi^2*sin(2*Phi)*cos(Theta)^2)/2 - Iyy*dPsi*dTheta*cos(2*
Phi)*cos(Theta) + Izz*dPsi*dTheta*cos(2*Phi)*cos(Theta);
20
21 CdE2=dPhi*(Ixx*dPsi*cos(Theta) - 2*Iyy*dTheta*cos(Phi)*sin(Phi) + 2*
Izz*dTheta*cos(Phi)*sin(Phi) + Iyy*dPsi*cos(Phi)^2*cos(Theta) -
Izz*dPsi*cos(Phi)^2*cos(Theta) - Iyy*dPsi*cos(Theta)*sin(Phi)^2 +
Izz*dPsi*cos(Theta)*sin(Phi)^2) - Ixx*dPsi^2*cos(Theta)*sin(
Theta) + Iyy*dPsi^2*cos(Theta)*sin(Phi)^2*sin(Theta) + Izz*dPsi^2*
cos(Phi)^2*cos(Theta)*sin(Theta);
22 CdE3=-dPhi*(Ixx*dTheta*cos(Theta) - Iyy*dTheta*cos(Phi)^2*cos(Theta) +
Izz*dTheta*cos(Phi)^2*cos(Theta) + Iyy*dTheta*cos(Theta)*sin(
Phi)^2 - Izz*dTheta*cos(Theta)*sin(Phi)^2 - 2*Iyy*dPsi*cos(Phi)*
cos(Theta)^2*sin(Phi) + 2*Izz*dPsi*cos(Phi)*cos(Theta)^2*sin(Phi)
) - Iyy*dTheta^2*cos(Phi)*sin(Phi)*sin(Theta) + Izz*dTheta^2*cos(
Phi)*sin(Phi)*sin(Theta) + 2*Ixx*dPsi*dTheta*cos(Theta)*sin(

```

```

Theta) - 2*Izz*dPsi*dTheta*cos(Phi)^2*cos(Theta)*sin(Theta) - 2*
Iyy*dPsi*dTheta*cos(Theta)*sin(Phi)^2*sin(Theta);
23 CdEta=[CdE1; CdE2; CdE3];
24
25
26 y = inv(J)*(TauB-CdEta);

```

En esta función se calcula el vector de momentos (\mathbf{TauB}), se calcula la actual matriz de cambio de base de velocidad angular a derivadas de ángulos de Euler (\mathbf{W}_η), se actualiza el tensor de inercias según este cambio y se obtiene el producto $\mathbf{C}(\boldsymbol{\eta}, \dot{\boldsymbol{\eta}}) \cdot \dot{\boldsymbol{\eta}}$. Finalmente devuelve las aceleraciones según se obtienen en la ecuación 27.

Los parámetros que se cargan en Matlab necesarios para definir este modelo son

```

1 % Declaracion simbolica de las variables
2
3 syms x dx ddx y dy ddy z dz ddz Phi dPhi ddPhi Theta dTheta ddTheta
4 Psi dPsi ddPsi
5 syms f1 f2 f3 f4
6 syms Ixx Iyy Izz
7 syms Ax Ay Az
8 syms k m g l b
9 l=0.165
10 Ixx=0.004
11 Iyy=0.004
12 Izz=0.008
13 m=0.85
14 b=1.2*10^-7
15 g=9.81
16
17 Ax=0
18 Ay=0
19 Az=0

```

*% Longitud de los brazos
% Producto de inercia en x
% Producto de inercia en y
% Producto de inercia en z
% Masa del Quadcopter
% Relacion Momento-Fuerza
% Aceleracion gravedad
% Friccion aire (no usado)*

4. Diseño del controlador

Con el modelo linealizado del sistema ya se tiene lo que emulará una respuesta verosímil y acorde a la realidad. Para que haga lo que uno espera se decide aplicar un control óptimo mediante un regulador LQR (Linear Quadratic Regulator).

4.1. Regulador Quadrático Lineal

Dado un sistema lineal de tiempo continuo definido por

$$\begin{aligned}\dot{\mathbf{X}} &= [\mathbf{A}] \cdot \mathbf{X} + [\mathbf{B}] \cdot \mathbf{U} \\ \mathbf{Y} &= [\mathbf{C}] \cdot \mathbf{X}\end{aligned}\quad (39)$$

se quiere operar este sistema mediante un coste mínimo según un criterio determinado. Este criterio se caracteriza por tener una función de costes \mathbf{J} que depende tanto de los estados como de las entradas del sistemas [8]

$$\mathbf{J} = \int_0^{\infty} (\mathbf{X}^T \mathbf{Q} \mathbf{X} + \mathbf{U}^T \mathbf{R} \mathbf{U} + 2\mathbf{X}^T \mathbf{N} \mathbf{U}) dt \quad (40)$$

y se definen los pesos \mathbf{Q} , \mathbf{R} y \mathbf{N} como matrices definidas positivas y simétricas, según la penalización que se le quiera dar a los estados (\mathbf{X}), acciones (\mathbf{U}) o el efecto de ambos (\mathbf{N}) respectivamente, tal que se minimice el coste \mathbf{J} . Se obtiene la ley de control $\mathbf{U} = -\mathbf{K} \cdot \mathbf{X}$ de

$$\mathbf{K} = \mathbf{R}^{-1} (\mathbf{B}^T \mathbf{S} + \mathbf{N}^T) \quad (41)$$

donde \mathbf{S} es la solución de la ecuación de Riccati asociada

$$\mathbf{A}^T \mathbf{S} + \mathbf{S} \mathbf{A} - (\mathbf{S} \mathbf{B} + \mathbf{N}) \mathbf{R}^{-1} (\mathbf{B}^T \mathbf{S} + \mathbf{N}^T) + \mathbf{Q} = -\dot{\mathbf{S}} \quad (42)$$

En este caso se supondrá que $\mathbf{N} = 0$ (ver [9]) se tendrá

$$\mathbf{J} = \int_0^{\infty} (\mathbf{X}^T \mathbf{Q} \mathbf{X} + \mathbf{U}^T \mathbf{R} \mathbf{U}) dt \quad (43)$$

y por consiguiente, la ecuación a resolver

$$\mathbf{A}^T \mathbf{S} + \mathbf{S} \mathbf{A} - \mathbf{S} \mathbf{B} \mathbf{R}^{-1} \mathbf{B}^T \mathbf{S} + \mathbf{Q} = 0 \quad (44)$$

Finalmente, de éste procedimiento se obtiene, además de las constantes de realimentación \mathbf{K} y la solución \mathbf{S} , los polos del sistema: $\mathbf{P} = \text{eig}(\mathbf{A} - \mathbf{B} \cdot \mathbf{K})$.

4.2. Obtención de la ley de control

Como se ha visto en la ecuación 34, las matrices que representan el sistema son

$$\begin{aligned}\mathbf{A} &= \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix} & \mathbf{B} &= \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & -l/I_{xx} & 0 & l/I_{xx} \\ 0 & 0 & 0 & 0 \\ -l/I_{yy} & 0 & l/I_{yy} & 0 \end{bmatrix} \\ \mathbf{C} &= \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}\end{aligned}\quad (45)$$

En el apartado anterior se ha visto que se deben asignar valores a \mathbf{Q} y \mathbf{R} , y como tanto \mathbf{X} como \mathbf{U} son vectores 4×1 , las matrices a asignar serán 4×4 . La matriz \mathbf{Q} se puede escribir como

$$\mathbf{Q} = 1 \cdot \mathbf{C}^T \mathbf{C} \quad (46)$$

por lo tanto

$$\mathbf{Q} = \begin{bmatrix} 1 & 0 \\ 0 & 0 \\ 0 & 1 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \quad (47)$$

Para el caso de \mathbf{R} se consideran toda un mismo peso para todas las entradas f_i y muy superior a los pesos de \mathbf{Q} :

$$\mathbf{R} = 100 \cdot I = \begin{bmatrix} 100 & 0 & 0 & 0 \\ 0 & 100 & 0 & 0 \\ 0 & 0 & 100 & 0 \\ 0 & 0 & 0 & 100 \end{bmatrix} \quad (48)$$

Se hayan \mathbf{K}, \mathbf{S} y \mathbf{E} mediante la toolbox de Matlab para calcular el control óptimo según el regulador LQR[10]:

`[K, S, e] = LQR(A, B, Q, R)`

Teniendo en cuenta que los parámetros del sistema físico obtenidos en el Anexo 4 son $I_{xx} = \dots$, $I_{yy} = \dots$ y $l = \dots$ se tiene que:

$$\begin{aligned} \mathbf{K} &= \dots & \mathbf{S} &= \dots \\ \mathbf{E} &= \dots \end{aligned} \quad (49)$$

4.3. Aplicación en el modelo

5. Implementación del control

6. Construcción del Quadcopter

El conjunt de peces que formen aquest aparell estan connectades entre sí segons la funció que realitzen. Esencialment la Raspberry Pi controla els motors segons les seïals que reb de l'IMU i el Receptor. Tot el conjunt és alimentat per una bateria LiPo i s'adapta el voltatge de 11.1V a 5V per mitjà d'un Regulador per tal d'alimentar a la Raspberry. Tots els components estan subjectats a una estructura (Frame) que també pateix les forces y moments.

6.1. Descripción de los componentes

Es descriu tot seguit cada component i el criteri de selecció que s'ha aplicat.

Raspberry Pi

Abreujat com a RPi, és un petit ordinador integrat en una sola placa (Single-Board Computer o SBC en anglès) del tamany d'una targeta de crèdit, és a dir, amb unes dimensions de 85.6cm x 53.98cm, desenvolupat per la Fundació Raspberry Pi amb l'intenció de promocionar les ciències computacionals a les escoles [2].



Figura 5: Raspberry Pi a utilitzar

S'ha optat per aquesta opció pel seu econòmic preu, la velocitat de processament i baix consum. A més, s'ha volgut ampliar els coneixements d'aquest petit monstre. En particular s'utilitza la segona revisió del model B:

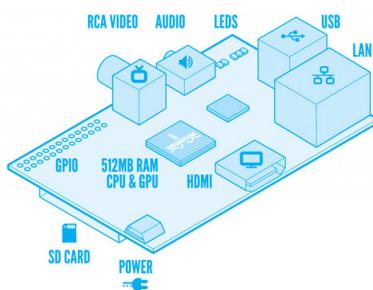


Figura 6: Conectors de la Raspberry Pi

Té un System-on-Chip (SoC) Broadcom BCM2835 amb un ARM1176JZF-S a 700 Mhz, una GPU VideoCore IV i 512 MB de memòria RAM. Disposa de dos ports USB, una sortida mini-jack 3.5mm, sortida d'audio/vídeo HDMI, una sortida RCA i un port RJ45 10/100 d'Ethernet.

L'alimentació es realitza per mitjà d'un mini USB a 5V/700mA, amb un consum de 3.5W. El sistema operatiu és un Raspbian, gravat en una targeta SD de 4GB. Disposa d'un conjunt de pins que permeten comunicació amb perifèrics de baix nivell UART, I2C, SPI i 8 pins de propòsit general (General Purpose Input Output o GPIO).

GY-521 MPU-6050

Es tracta d'una Unitat de Mesura Inercial (IMU en anglès) que integra en un mateix encapsulat de $4x4x0,9mm$ un acceleròmetre i un giròscop, ambdós de 3 eixos. Disposa dun conversor ADC de 16 bits per a cada eix i es comunica mitjançant un protocol de comunicació *I2C*. S'ha optat per utilitzar aquest dispositiu pel seu baix cost i la fàcil comunicació que comporta amb la RPi.



Figura 7: IMU MPU-6050

Com a característiques dels sensors: el giròscop té un rang de $\pm 250, \pm 500, \pm 1000, \pm 2000$ graus/-segon, i l'acceleròmetre de $\pm 2g, \pm 4g, \pm 8g, 16g$. La tensió d'alimentació és del rang de $2,375V - 3,46V$ i cap la possibilitat d'utilitzar un mòdul *DMP* (Digital Motion Processor), però s'ha decidit implementar un filtre de Kalman per llegint les dades en cru (raw) de la cua *FIFO* del sensor.

Emissor-Receptor

Amb aquest parell de components es transmet la consigna generada des del transmissor cap al receptor mitjançant ones de radio. El model que s'utilitza és el *Turnigy5X5ChMini*, per què és un model fàcil d'utilitzar i econòmic. Les especificacions tècniques més rellevants són:



Figura 8: Emissora i Receptor

El transmissor té unes dimensions de $156x152x50mm$, un pes de 265g i s'alimenta a 6V (4 bateries AA). El receptor té unes mides de $33,5x20,5x13mm$ i s'alimenta a 4.8-6V. Disposa de 5 canals de radio, amb transmissió segura a 2.4GHz amb el mètode FHSS. Pot configurar-se per treballar amb dos modes (mode1-mode2).

La senyal que es reb per cada canal en el receptor és de *PWM* de $50Hz$ amb un Duty que varia de $1ms$ a $2ms$.

Bateria LiPo

Per a alimentar a tot el conjunt s'utilitza una LiPo *Turnigy2200mAh3S1P25C*. Per tant, és capaç d'entregar 2.2A durant una hora, i com que la capacitat és de 25C, la descàrrega pot ser de $2,2 * 25 = 55A$ amb un pic de descàrrega de 35C, és a dir, amb un pic de $2,2 * 35 = 77A$ durant 10 segons. Aquesta bateria està formada per tres cel·les que proporcionen un voltatge total d'uns 11,1V:



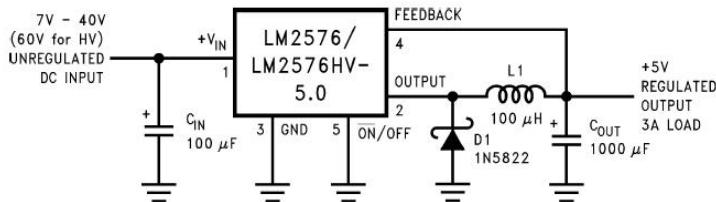
El conector de càrrega és el *JST - XH* i el de descàrrega el *XT60*. Per a tenir en compte amb per al quadcopter és interessant saber que pesa 188g i respecte a l'estructura (frame) que les seves dimensions són de 105x33x24mm.

Regulador Step-Down

Per adaptar la tensió de 11.1V de la bateria LiPo a 5V per alimentar la Raspberry Pi, és necessari un regulador. En aquest cas es té l'Interruptor-Mode Màxim BEC LM2576S, que proporciona fins a 3A de corrent



L'esquema del component [6]:



Motors Brushless

Els actuadors que s'utilitzen són motors brushless (BL-DC). En particular el Turnigy 2213 20turn 1050kv 19A Outrunner. Es tracta de màquines síncrones d'imans permanents, en competes d'utilitzar electroimants, on el camp magnètic està uniformement repartit a l'entreferro. S'anomenen també de força electromotriu (FEM) trapezoidal perquè a velocitat constant la FEM té aquesta forma. Per a controlarl-lo s'utilitzar un inversor (inverter), la commutació deixa de ser mecànica i per tant l'armadura de la màquina pot ser l'estator. Això permet arribar a voltatges d'ús i velocitats de funcionament més alts. Les seves especificacions més rellevants són:

- Kv: 1050rpm/V
- Corrent de treball: 6A 16A
- Corrent de pic: 19A
- Pes: 56g
- Dimensions: 27.6 x 32mm
- Mida de l'eix: 3.175mm

El primer paràmetre fa referència a quants rpm's funcionarà el motor per Volt aplicat. A aquest valor se li aplica un percentatge $\%NLS$ (Percentatge de Velocitat sense Càrrega, típicament de 70%) per a quantes revolucions per minut funcionarà:

$$Kv \cdot V \cdot \%NLS = 1050 \cdot \frac{rpm}{V} \cdot 0,70 = 8085 \text{ rpm} \quad (50)$$

Aquesta velocitat és teòrica, i com que no hi ha un llaç de control sobre aquesta, no es tindrà en compte. En canvi, s'atacarà al motor amb la senyal de PWM que s'envia al Variador que el controla. La relació entre la senyal de PWM i la força, així com el consum que es desenvolupa es troben als Annexos.

Variador ESC

Para controlar un motor brushless la etapa de potencia a utilitzar es un ESC (Electronic Speed Controller) que es un variador que se compone de un inversor trifásico: El modelo a

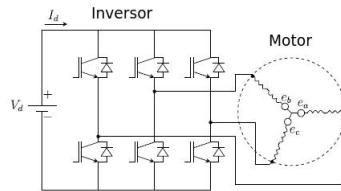


Figura 9: Esquema de un Variador como inversor trifásico

utilitzar es el Turnigy AE-20A Brushless ESC. El uso de este tipo se justifica con el amperaje que soporta, en cuanto los motores consumen un pico de hasta 19A, que es menor al màxim de la corriente de este Variador (como límite se puede quemar el componente si se hacen circular 25A durante 10 segundos). Se debe asegurar la compatibilidad con la batería también, ya que soporta tensiones de entrada correspondientes a Baterías LiPo de entre 2 y 4 celdas. Incluye un circuito BEC (Battery Eliminator Circuit) para poder alimentar también el receptor de 2A de máxima corriente. La máxima velocidad de funcionamiento del motor depende de sus polos: si es de 2 polos será de 210000 rpm, con 6 polos de 70000 rpm y de 12 polos a 35000 rpm.

Tanto el peso de 19g, como las dimensiones de 50x26x12mm son parámetros a tener en cuenta en el montaje, en parte a que se hubican en los brazos del Quadcopter, aumentando la inercia por eje e influyendo considerablemente en el modelo. Antes de usar cada Variador se ha tenido que programar para que éste responda con un mejor rendimiento al caso en cuestión.

Throttle stick position	Low	Medium	High
Menu			
1. Brake setting	• off	soft	hard
2. Timing setting	Low	• Medium	High
3. Battery Protection Voltage Threshold	High cut-off threshold	• Medium cut-off threshold	—
4. Plane Mode	• Fixed-wing aircraft	Helicopter mode 1 (with Soft start)	Helicopter mode 2 (Super-Soft+Governor Mode)
5.Throttle response speed setting	• normal	Medium	High

Figura 10: Tabla de Set up de un ESC

Para los 5 parámetros a configurar hay tres opciones, y las marcadas con un punto son las que se han aplicado.

Marco

Todos los elementos que componen el Quadcopter se sujetan en una estructura que mantiene la integridad y rigidez necesarias para el vuelo.

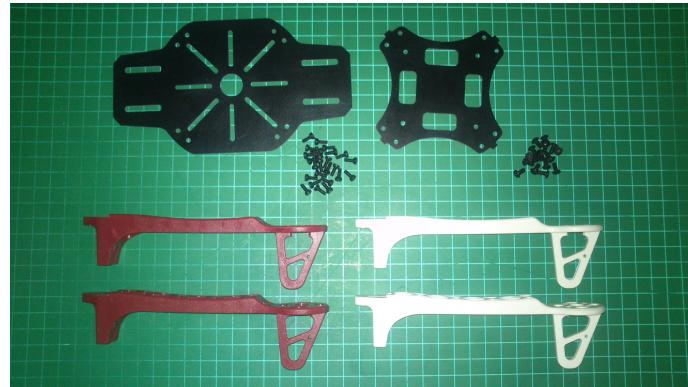


Figura 11: Piezas que componen la estructura del Quadcopter

La mayor parte del peso se concentra en el centro de la estructura e interesa que sea lo más ligera posible sin comprometer su integridad, de tal manera que los motores, situados en las puntas de cada brazo, deban compensar el menor peso posible. En este caso, tiene un peso de 190g, y las piezas están hechas de fibra de vidrio las piezas centrales (de color negro en la imagen) y de Nylon los brazos (rojo y blanco en la imagen). La distancia entre dos motores de una diagonal es de 33cm.

Hélices

En girar el motor, las aspas solidarias a su eje utilizan la energía cinética del rotor para proporcionar sustentación propulsando aire en sentido contrario. El tipo de aspa condiciona el rendimiento que se obtenga, así que se utilizan las que optimizan la fuerza de propulsión respecto al consumo de corriente. El modelo de hélice utilizado es el 8x4



Figura 12: Par de hélices 8x4 simétricas

Los parámetros que definen una hélice son su Longitud (X), es decir, diámetro de superficie recorrida por las aspas en pulgadas, y el Pitch (Y), que es la distancia que se avanzaría la hélice en una revolución de moverse en una masa blanda. Se representan en el siguiente gráfico:

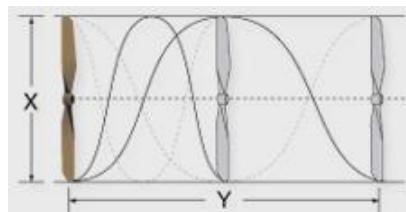
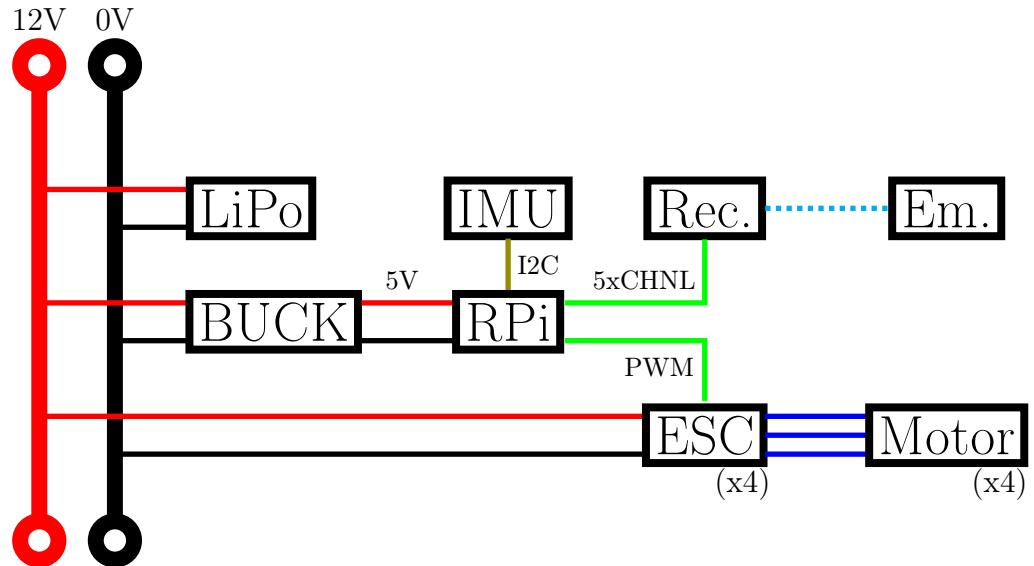


Figura 13: Representación de la Longitud y Pitch de una Hélice

6.2. Montaje



7. Análisis económico

8. Estudio del impacto ambiental

9. Conclusiones

Futuras aplicaciones

Referencias

- [1] Teppo Luukkonen (August 22,2011). *Modelling and control of quadcopter*
- [2] Wikipedia de la Raspberry Pi: http://en.wikipedia.org/wiki/Raspberry_Pi
- [3] Acelerómetro y giroscopio MPU-6050 para Arduino: <http://playground.arduino.cc/Main/MPU-6050#.UzhsVCK9jb4>
- [4] MPU-6050. Especificación del producto: <http://www.invensense.com/mems/gyro/documents/PS-MPU-6000A-00v3.4.pdf>
- [5] MPU-6050. Mapa de Registros y descripciones: <http://www.invensense.com/mems/gyro/documents/RM-MPU-6000A-00v4.2.pdf>
- [6] LM2576S Datasheet: <http://www.ti.com/lit/ds/symlink/lm2576.pdf>
- [7] Librería wiringPi: <http://wiringpi.com/>
- [8] Artículo sobre LQR de Wikipedia: <http://en.wikipedia.org/wiki/Linear-quadratic-regulator>
- [9] Documentación de la función LQR de Matlab: <http://www.mathworks.es/help/control/ref/lqr.html;jsessionid=8f737c267ee85a89cc2544a09a44>
- [10] Toolbox de Matlab del LQR : <http://www.mathworks.es/es/help/control/ref/lqr.html>

Agradecimientos

ANEXOS

Anexo 1: Obtención del vector de fuerzas

Para trabajar con la dinámica del sistema se obtiene el vector de fuerzas a partir del Lagrangiano:

```

1 clear all
2 clc
3
4 %%Para usar la funcion Laplace en el archivo Laplace.m se deben
5 % declarar las variables de esta manera:
6
7 syms x dx ddx y dy ddy z dz ddz Phi dPhi ddPhi Theta dTheta ddTheta
8 Psi dPsi ddPsi
9 syms f1 f2 f3 f4 m1 m2 m3 m4
10 %%syms w1 w2 w3 w4
11 syms Ixx Iyy Izz
12 syms Ax Ay Az           %No se tienen en cuenta coeficientes de
13 %friccion
14 syms k m g l b
15
16 l=0.165                  %longitud de los brazos
17 Ixx=0.004                 %Momento de inercia del eje x
18 Iyy=0.004                 %Momento de Inercia del eje y
19 Izz=0.008                 %Momento de Inercia del eje z
20 m=0.85                   %Peso del Quadcopter
21 g=9.81                    %gravedad
22
23 %%El vector de variables que se usaran
24 v=[x dx ddx y dy ddy z dz ddz Phi dPhi ddPhi Theta dTheta ddTheta
25 %Psi dPsi ddPsi]
26
27 Xi=[x; y; z]
28 dXi=[dx; dy; dz]
29
30 Eta=[Phi; Theta; Psi]
31 dEta=[dPhi; dTheta; dPsi]
32
33 Q=[Xi; Eta]
34 dQ=[dXi; dEta]
35
36 II=[[Ixx 0 0];
37 [0 Iyy 0];
38 [0 0 Izz]]
39
40 A=[[Ax 0 0];
41 [0 Ay 0];
42 [0 0 Az]]
43
44 R=[[cos(Psi)*cos(Theta) cos(Psi)*sin(Theta)*sin(Phi)-sin(Psi)*cos(Phi)
      cos(Psi)*sin(Theta)*cos(Phi)+sin(Psi)*sin(Phi)];
      [sin(Psi)*cos(Theta) sin(Psi)*sin(Theta)*sin(Phi)+cos(Psi)*cos(Phi)
      sin(Psi)*sin(Theta)*cos(Phi)-cos(Psi)*sin(Phi)];
      [-sin(Theta) cos(Theta)*sin(Phi) cos(Theta)*cos(Phi)]]

```

```

45 Weta=[[1 0 -sin(Theta) ];
46 [0 cos(Phi) cos(Theta)*sin(Phi) ];
47 [0 -sin(Phi) cos(Theta)*cos(Phi) ]]
48
49 J=transpose(Weta)*II*Weta
50
51 L=m/2*transpose(dXi)*dXi+(1/2)*transpose(dEta)*transpose(Weta)*II*
52 Weta*dEta-m*g*z
53 %% Aqui se obtiene con el Lagrangiano el vector de fuerzas F segun
54 %% F=(d/dt)(dp(L)/dp(dq))-d(L)/dq
55
56 F=Lagrange(L,v)
57
58 %% Derivando a mano se obtiene que TauB=J*ddEta+CdEta
59 %% Y como TauB=[F(4); F(5); F(6)], se tiene que el termino de
59 %% Coriolis es:
60
61 CdEta=[F(4); F(5); F(6)]-J*[ddPhi; ddTheta; ddPsi]
62
63 %% Hasta se tiene el producto Coriolis(Eta,dEta)*dEta. Pero como
63 %% matlab
64 %% no elimina las aceleraciones se hace a mano, y queda:
65
66 CdEta = [(Iyy*dTheta^2*sin(2*Phi))/2 - (Izz*dTheta^2*sin(2*Phi))/2 -
66 Ixx*dPsi*dTheta*cos(Theta) - (Iyy*dPsi^2*sin(2*Phi)*cos(Theta)
66 ^2)/2 + (Izz*dPsi^2*sin(2*Phi)*cos(Theta)^2)/2 - Iyy*dPsi*dTheta*
66 cos(2*Phi)*cos(Theta) + Izz*dPsi*dTheta*cos(2*Phi)*cos(Theta);
67 dPhi*(Ixx*dPsi*cos(Theta) - 2*Iyy*dTheta*cos(Phi)*sin(Phi)
67 + 2*Izz*dTheta*cos(Phi)*sin(Phi) + Iyy*dPsi*cos(Phi)
67 ^2*cos(Theta) - Izz*dPsi*cos(Phi)^2*cos(Theta) - Iyy*
67 dPsi*cos(Theta)*sin(Phi)^2 + Izz*dPsi*cos(Theta)*sin(
67 Phi)^2) - Ixx*dPsi^2*cos(Theta)*sin(Theta) + Iyy*dPsi
67 ^2*cos(Theta)*sin(Phi)^2 + Izz*dPsi^2*cos(Phi)^2*cos(Theta);
68 -dPhi*(Ixx*dTheta*cos(Theta) - Iyy*dTheta*cos(Phi)^2*cos(
68 Theta) + Izz*dTheta*cos(Phi)^2*cos(Theta) + Iyy*dTheta*
68 cos(Theta)*sin(Phi)^2 - Izz*dTheta*cos(Theta)*sin(Phi)
68 ^2 - 2*Iyy*dPsi*cos(Phi)*cos(Theta)^2*sin(Phi) + 2*Izz*
68 dPsi*cos(Phi)*cos(Theta)^2*sin(Phi) - Iyy*dTheta^2*cos(
68 Phi)*sin(Phi)*sin(Theta) + Izz*dTheta^2*cos(Phi)*sin(
68 Phi)*sin(Theta) + 2*Ixx*dPsi*dTheta*cos(Theta)*sin(
68 Theta) - 2*Izz*dPsi*dTheta*cos(Phi)^2*cos(Theta)*sin(
68 Theta) - 2*Iyy*dPsi*dTheta*cos(Theta)*sin(Phi)^2*sin(
68 Theta)]]
69
70 T=f1+f2+f3+f4 % Fuerza total (Total Thrust)
71 TB=[0; 0; T] % Thrust en la referencia del
71 % Quadcopter
72 TauB=[1*(f4-f2);
73 1*(f3-f1);
74 % (m1+m2+m3+m4)/
75 b*(f1-f2+f3-f4)]

```

Anexo 2:Preparación de la Raspberry Pi

Para poner a punto la RPi se instala un Raspbian y se configura para que es pueda comunicar con la placa MPU-6050.

Instalación Raspbian

Habiendo introducido la tarjeta SD en un lector adecuado, se detecta en una terminal mediante la orden `df -h`. Suponiendo que haya estado gravada anteriormente:

```
/dev/sdb1      56M   19M   38M  33% /media/boot
/dev/sdb2     1,8G  1,4G  266M 85% /media/9c7e2035-df9b-490b-977b-d60f2170889d
```

Se deben desmontar las dos particiones, tanto la `/dev/sdb1` como la `/dev/sdb2`:

```
umount /dev/sdb1
umount /dev/sdb2
```

Descargada la imagen a instalar, por ejemplo en el Escritorio, se procede con:

```
sudo dd bs=4M if=~/Escritorio/2013-07-26-wheezy-raspbian.img of=/dev/sdb
```

Pasados unos minutos ya se tiene la SD grabada con el sistema operativo Raspbian.

Configuración de la Raspberry

Para poder comunicar la RPi con la IMU MPU-6050 por I2C es necesario configurar el sistema. Primero es necesario instalar los drivers mas relevantes. Del archivo

```
sudo nano /etc/modules
```

se añaden las siguientes dos líneas al final del archivo:

```
i2c-bcm2708
i2c-dev
```

En el archivo blacklist:

```
sudo vi /etc/modprobe.d/raspi-blacklist.conf
```

las siguientes dos líneas deben empezar con un signo # (de comentario):

```
#blacklist spi-bcm2708
#blacklist i2c-bcm2708
```

Para conectar el sensor, es necesario seguir primero el conexionado:

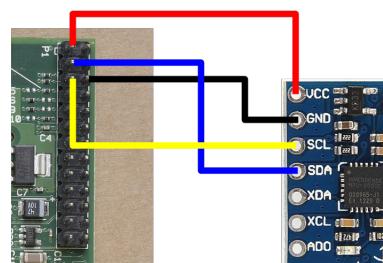


Figura 14: Conexión por I2C entre MPU-6050 y RPi

Es decir

- Pin 1-3.3V se conecta a VCC.

- Pin 3-SDA se conecta a SDA
- Pin 5-SCL se conecta a SCL
- Pin 6-Ground se conecta a GND

Es necesario instalar el paquete *i2c-tools*:

```
sudo apt-get install i2c-tools
```

Para ver el sensor se escribe:

```
sudo i2cdetect -y 1
```

El resultado es

```
pi@raspberrypi ~ $ sudo i2cdetect -y 1
      0  1  2  3  4  5  6  7  8  9  a  b  c  d  e  f
00: -- - - - - - - - - - - - - - - - - - - - - - -
10: -- - - - - - - - - - - - - - - - - - - - - -
20: -- - - - - - - - - - - - - - - - - - - - - -
30: -- - - - - - - - - - - - - - - - - - - - - -
40: -- - - - - - - - - - - - - - - - - - - - - -
50: -- - - - - - - - - - - - - - - - - - - - - -
60: -- - - - - - - - - - - - - - - - - - - - 68 - -
70: -- - - - - - - - - - - - - - - - - - - - - -
```

Figura 15: Detección del sensor MPU-6050

que es el dispositivo que corresponde al MPU-6050.

Librería wiringPi

Se utiliza esta librería [7] como a interfaz del GPIO (General Purpose Input Output). Permite, por tanto, leer las consignas y sensores y generar las acciones de control. La instalación y descripción de la librería esta explicado en la página web incluida en la bibliografía.

Anexo 3: Filtro de Kalman

Anexo 4: Identificación de los parámetros del Quad

Inercias

motores

Para poder controlar la planta es necesario saber cómo responden los actuadores para realizar un control preciso. De éstos se deben conocer qué fuerza y momento proporcionan, así como el consumo de energía, dado un PWM. Para obtener una curva de la fuerza respecto del PWM se pesa con una balanza la acción del motor. La rudimentaria bancada que se tiene para tal objetivo es un codo articulado en el eje de tal manera que la distancia del eje del motor al fulcro y de éste a la balanza sea la misma (en este caso de 12 cm):

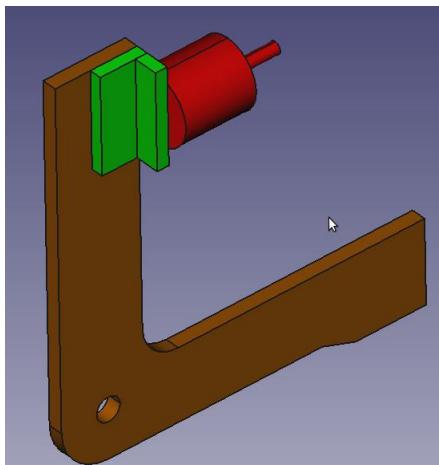


Figura 16: Bancada para el estudio del motor

Las partes están hechas de madera (parte marrón) y plástico ABS (parte verde). Toda la estructura se sujeta a unas maderas:



Figura 17: Montaje para el estudio del motor

Se transmite al ESC a qué velocidad se debe hacer funcionar el motor des de la Raspberry Pi por medio de una señal PWM. El programa que se ejecuta para hacer las pruebas es:

```

1 #include <stdio.h>
2 #include <termios.h>
3 #include <wiringPi.h>
4 #include <unistd.h>
5
6 #define PIN 2 // pin 13

```

```

7  int main(void){
8
9      int i ,p;
10     wiringPiSetup () ;
11
12     pinMode(PIN ,OUTPUT) ;
13     digitalWrite(PIN ,LOW) ;
14
15     for ( i=0;i <=100;i +=5){
16         for (p=0;p <250;p++){
17             digitalWrite(PIN ,HIGH) ;
18             delayMicroseconds(950+i *10) ;
19             digitalWrite(PIN ,LOW) ;
20             delayMicroseconds(19050 - i *10) ;
21
22             printf( "PWM( i ) : %d \t h: %d \n" ,i ,p) ;
23         }
24     }
25     return 0;
26 }
27 }
```

Se obtiene una lectura de fuerza y de intensidad por cada incremento del 0.25% del PWM enviado al Variador. Las lecturas que se obtienen son:

PWM(%)	Fuerza(g)	Intensidad(A)
5	0	0
5,25	0,25	13
5,5	0,5	38
5,75	0,9	74
6	1,38	98
6,25	1,8	128
6,5	2,4	158
6,75	2,9	191
7	3,5	215
7,25	4	250
7,5	4,6	277
7,75	5,5	318
8	6,5	360
8,25	8	400
8,5	9,15	450

Graficado:

En el eje de las abscisas el PWM(%) es entre el 5% y el 10% de la señal enviada, que corresponden al mínimo y máximo del motor. Para un PWM elevado la fuerza era demasiado grande y no se prevé que se haga funcionar al sistema en semejantes condiciones de trabajo, no se estudia.

Se ve una relación proporcional del PWM con la fuerza generada. Se considera que una regresión lineal será suficiente. Se obtiene que la fuerza $f(PWM)$ del motor será:

$$f = 32,5758 \cdot PWM(\%) - 32,1758$$

con un coeficiente de determinación en la regresión de $R^2 = 0,9929$. Este resultado facilita el problema en ser una relación suficiente sencilla.

M(g)

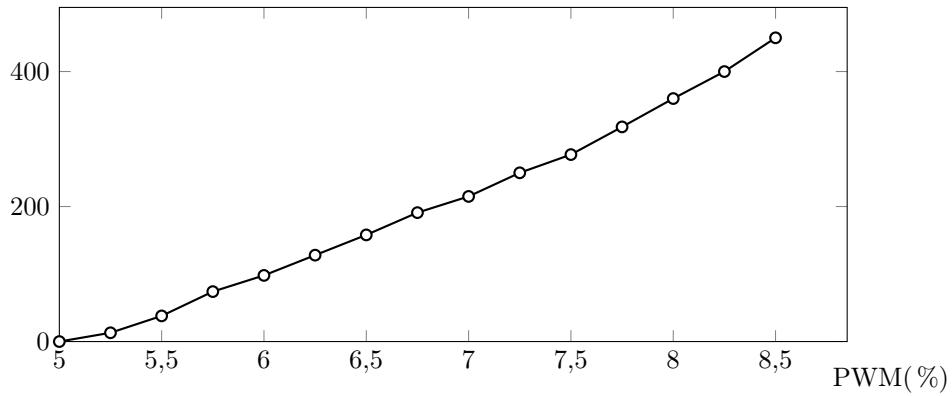
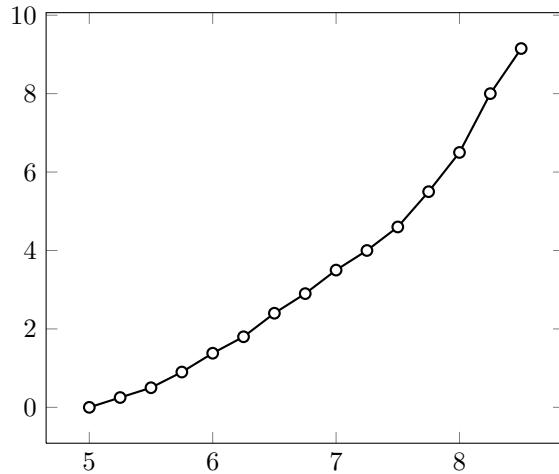


Figura 18: Relación entre la fuerza (en gramos) y el PWM

Para la intensidad se tiene una relación tan amigable como en el caso anterior, pero como no sera caso de estudio,solo se verá una aproximación de lo que puede llegar a consumir cada motor para asegurar una autonomía de vuelo no irrisoria. Resulta



Como en el punto de equilibrio cada motor soporta una cuarta parte del peso, el PWM debería corresponder al de una fuerza de $m_{Quadcopter} \cdot g / 4 \approx 0,225\text{kg}$. El consumo correspondiente es de 3,5A aproximadamente. Entre los cuatro motores se consumirán $3,5 \cdot 4 = 14\text{A}$. Como que la batería es de 2200mAh se supone una autonomía de 7 minutos aproximadamente.