



**Universidad**  
Zaragoza

## Trabajo Fin de Grado

Modelado, simulación y control de cuadricópteros  
Modelling, control and simulation of quadcopters

Autor

Enrique Cañada Panea

Director

Enrique Teruel Doñate

Escuela de Ingeniería y Arquitectura  
Departamento de Informática e Ingeniería de Sistemas  
Grado en Ingeniería Electrónica y Automática

Año  
2020

# Modelado, simulación y control de cuadricópteros

## Resumen

En este trabajo se realiza el modelado, el control y la simulación de un cuadricóptero. Para ello se utiliza el programa de simulación Modelica.

Primero se desarrolla un modelo completo del movimiento y las fuerzas que actúan sobre el cuadricóptero, es decir los modelos cinemático y dinámico del mismo. Después se elaboran distintos modelos simplificados para poder llevar a cabo el cálculo de los diferentes tipos de controles realizados, para lo cual se ha recurrido a métodos de linealización de un modelo fuertemente no lineal.

Estos controles incluyen control de velocidad angular de los rotores, control de orientación y control de posición del cuadrirrotor, desarrollados todos ellos con técnicas clásicas de función de transferencia.

Todos los controles se han elaborado en el mundo discreto para hacer posible su implementación mediante algoritmos en dispositivos digitales, como puede ser un ordenador o un microcontrolador.

## Tabla de contenido

1.	Introducción .....	3
2.	Funcionamiento y modelado del cuadricóptero .....	5
2.1.	Funcionamiento .....	5
2.2.	Modelado.....	7
2.2.1.	Modelo cinemático.....	7
2.2.2.	Modelo dinámico.....	10
2.2.3.	Modelo de los motores .....	14
2.3.	Entradas y salidas.....	16
2.4.	Verificación del modelo .....	16
3.	Control .....	23
3.1.	Control de velocidad angular de los rotores .....	24
3.1.1.	Control diseñado por emulación .....	26
3.1.2.	Control deadbeat.....	29
3.1.3.	Comparación de ambos métodos .....	31
3.2.	Control de orientación .....	34
3.2.1.	Realimentación taquimétrica .....	37
3.2.2.	Control deadbeat.....	39
3.2.3.	Comparación de ambos métodos .....	41
3.3.	Control de posición .....	47
3.4.	Generador de trayectorias.....	48
4.	Conclusiones.....	54
5.	Referencias .....	56
6.	Tabla de figuras .....	57

## 1. Introducción

Un cuadricóptero o cuadrirrotor es un vehículo aéreo no tripulado impulsado por cuatro hélices, es decir, un helicóptero con cuatro rotores. A lo largo de las últimas décadas se han desarrollado múltiples formas de modelar y controlar estos dispositivos, pero todas tienen algo en común: al tener solamente cuatro rotores (y por tanto motores) se dispone únicamente de cuatro variables a las que se tiene acceso y poder directamente, por lo que para poder controlar su movimiento se debe primero observar y analizar de qué manera la combinación esas cuatro variables (las tensiones aplicadas a los motores de las hélices) afecta al movimiento general del cuadrirrotor (traslación y giro), para después ingeniar una serie de controles que permitan obtener un comportamiento deseado en el movimiento general del vehículo mediante la manipulación única de aquellas cuatro variables.

El objetivo general del presente trabajo es, por tanto, el de desarrollar primeramente el modelo matemático de la física de un cuadricóptero, con el programa Modelica, combinando sus diferentes componentes (motores, hélices, chasis y control), para disponer de manera aproximada de su comportamiento, para simular dicho comportamiento y posteriormente diseñar y probar su control automático. Este objetivo principal se podría descomponer en varios objetivos específicos:

- Realizar un modelo lo suficientemente fiel a la realidad para que se asemeje lo más posible a un cuadricóptero real, pero a la vez lo suficientemente sencillo para poder centrar mayor esfuerzo en la parte del control.
- Llevar a cabo distintas simulaciones para verificar dicho modelo
- Desarrollar distintos controles usando técnicas diferentes para poder comparar su poder y eficacia
- Realizar dichos controles en tiempo discreto para una posible posterior implementación en un microcontrolador
- Por último, pero no por ello menos importante, profundizar en el aprendizaje de controladores discretos

En cuanto a la metodología, tal como se ha dicho, en este trabajo se ha dado un mayor enfoque y desarrollo a la parte del control del sistema, realizando modelos sencillos de la parte física del cuadricóptero. La principal herramienta que se ha utilizado es el programa Modelica para diseñar, simular y verificar el modelado del sistema, y programar y probar diferentes algoritmos de control, desde el bajo nivel hasta el alto nivel - como el seguimiento de trayectorias. No obstante, se han usado además otras herramientas como MATLAB, gestores bibliográficos como Mendeley, manuales del programa Modelica como [1] o [2] y la bibliografía citada al final de esta memoria.

En referencia al cronograma, el trabajo se descompone en las siguientes tareas realizadas: estudio y aprendizaje de la materia y de las herramientas a utilizar; diseño y verificación de los modelos; análisis, diseño y programación de los controladores; diseño de experimentos; análisis de resultados y elaboración de la memoria.

De manera general, en la parte del modelado del cuadricóptero, este trabajo se ha apoyado principalmente en el trabajo realizado en [3] debido a su sencillez y claridad, pero también en [4], [5], [6], [7], [8] y [9] para completarlo y tener una comprensión mayor del mismo.

Además, se han elaborado distintos modelos simplificados para poder llevar a cabo el cálculo de los distintos tipos de controles realizados, para lo cual se ha recurrido a métodos de linealización de un modelo fuertemente no lineal. Para parte de la linealización nos hemos ayudado de los trabajos de [10] y [11].

En cuanto a los distintos controles, este trabajo se apoya principalmente en [4] para elaborar el control de posición y la generación de trayectorias.

En las próximas secciones del presente trabajo se desarrollará el contenido del mismo: el funcionamiento cualitativo y el modelado cuantitativo del cuadricóptero en la sección 2, que incluye modelo cinemático, modelo dinámico y modelo de los motores, así como un breve análisis de las entradas y las salidas de nuestro sistema. También se realizará la verificación del modelo mediante una serie de experimentos.

En la sección 3 se explica la realización de modelos linealizados y más simplificados, se detallan todos los controles – de velocidad angular de los motores, de orientación y de posición del cuadricóptero – y se explica la elaboración de las generaciones de trayectorias.

A la par se realiza una comparación de los resultados obtenidos de distintos métodos de control: PID discretizado por emulación, deadbeat y por realimentación taquimétrica, evidenciando las ventajas y las limitaciones de cada uno de ellos.

Por último, se dan las conclusiones pertinentes y las futuras líneas de trabajo a seguir en esta temática.

## 2. Funcionamiento y modelado del cuadricóptero

### 2.1. Funcionamiento

El funcionamiento del cuadricóptero se basa en el giro de cuatro hélices para llevar a cabo todo el movimiento. Este movimiento de todo el cuerpo del vehículo consiste en giros en torno a sus tres ejes y traslación hacia arriba (arriba visto desde la perspectiva del robot), que combinados resultan además en traslaciones horizontales.

Para obtener este movimiento disponemos de cuatro rotores que se pueden encontrar en disposición de equis o disposición de cruz, tal como podemos ver en la Figura 1:

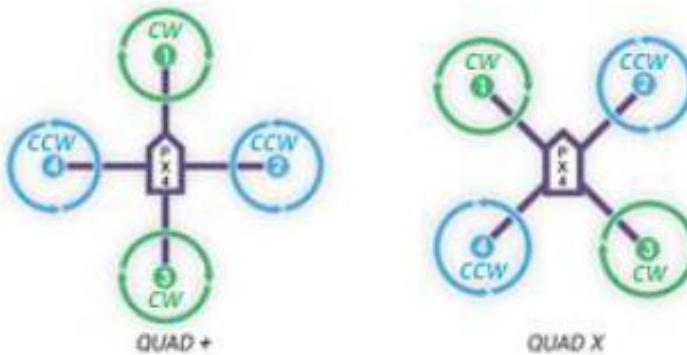


Figura 1. Rotores en cruz y en equis. Vista de planta. Tomado de [8].

En nuestro caso, consideramos que tienen disposición en cruz. El giro de estas hélices crea una fuerza de empuje que va en la dirección perpendicular al plano del robot y hacia arriba, por lo que ayuda a sustentarse a nuestro cuadricóptero, venciendo la acción de la gravedad.

Tomando como principio esta fuerza que crean los rotores, podemos deducir que, aumentando la velocidad de uno de ellos (por ejemplo, el número 1), tendremos más empuje en esa hélice y por tanto habrá un giro del robot alrededor del eje perpendicular a la línea que une ese rotor con más velocidad al centro de la cruz (el eje que une los rotores 2 y 4), giro con sentido positivo en torno al rotor 2 siguiendo la regla del sacacorchos. Como es obvio, este efecto también se puede conseguir disminuyendo la velocidad de la hélice contraria a la anterior (la 3). De la misma manera, podremos obtener un giro alrededor del eje perpendicular al anterior (ahora el eje que une los rotores 1 y 3) aumentando la velocidad del rotor 2 (o disminuyendo la del número 4) para un sentido de giro positivo hacia el rotor 3, o viceversa. Cabe destacar que con el cuadricóptero girado en torno a uno de los anteriores ejes, debemos aumentar la

velocidad total de los rotores, ya que el empuje vertical que ejercen ya no está alineado con la gravedad y por tanto su proyección sobre el eje vertical del mundo será menor que antes del giro.

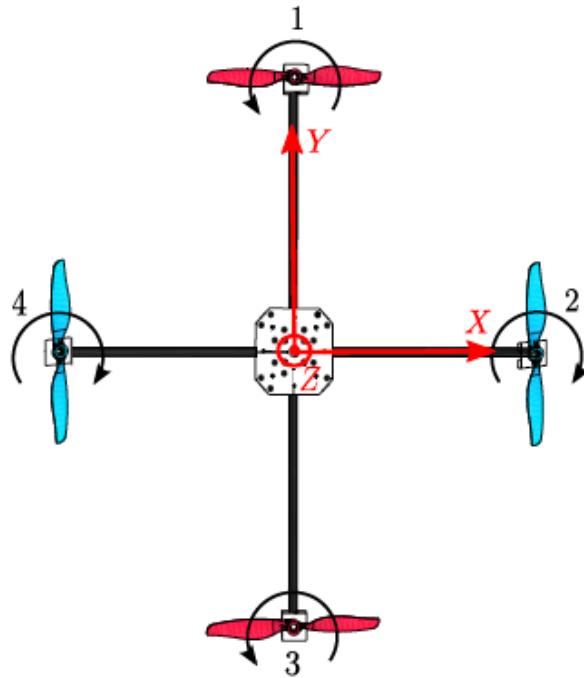


Figura 2. Giro de los rotores y definición de los ejes respecto de los números de los rotores. Tomado de [3].

Tenemos además que dos rotores opuestos giran en el mismo sentido, pero en sentido contrario respecto de los otros dos rotores, tal como se puede apreciar en la Figura 2. Esto se debe a que, por conservación del momento angular, cuando una hélice gira, crea un momento en torno al eje perpendicular al plano del robot para compensar dicho giro. Este momento tendrá sentido opuesto al giro de la hélice. Por lo cual, si todos los rotores giraran en la misma dirección tendríamos un momento constante haciendo girar al robot alrededor del eje Z de la figura. Disponiéndolos como aparece en la Figura 2, anulamos los momentos entre sí, al menos parcialmente.

Ya hemos explicado pues cómo obtener cada uno de los giros alrededor de los tres ejes del cuadricóptero, así como una traslación en el eje Z de la referencia del robot (o por lo menos una sustentación que contrarreste la acción de la gravedad). Se puede deducir a partir de lo anterior que ejerciendo una acción conjunta de giro alrededor del eje X o Y de la Figura 2 y empuje hacia arriba en el eje Z, conseguiremos una traslación negativa en el eje Y o positiva en el eje X, respectivamente.

Ya tenemos de esta manera declarada la forma cualitativa de llevar al robot a un movimiento y por tanto a una posición deseada.

Expliquemos esto de manera más extensa, rigurosa y cuantitativa en la siguiente sección, basándonos en [3].

## 2.2. Modelado

### 2.2.1. Modelo cinemático

Primero, debido a los distintos giros que realiza el robot, debemos definir dos sistemas de referencia o de coordenadas, como para la mayoría de robots: la referencia global o del mundo y la referencia local o del robot. De esta manera, tendremos dos espacios de trabajo: uno en el que “nos subimos” al robot y lo vemos todo desde su perspectiva, y otro en el que desde una posición exterior observamos el movimiento global del vehículo. Ambos sistemas de coordenadas siguen la regla de la mano derecha de los sistemas de referencia cartesianos.

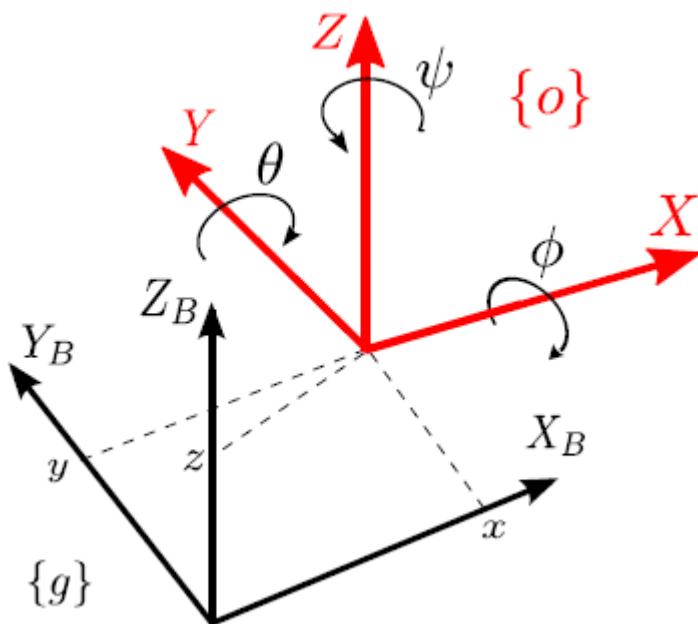


Figura 3. Ejes de coordenadas globales y locales. Tomado de [3]

De esta forma, podemos definir ya la posición del centro de masas de nuestro robot con  $x$ ,  $y$  y  $z$  respecto de la referencia global. También podemos definir la orientación o el giro (que sigue la regla del sacacorchos) en torno a los ejes  $X$ ,  $Y$  y  $Z$  de la referencia local como  $\varphi$  (phi),  $\vartheta$  (theta) y  $\psi$  (psi), respectivamente. Podemos representar las magnitudes anteriores, respectivamente, con los vectores:  $\mathbf{x} = [x \ y \ z]^T$  y  $\boldsymbol{\Psi} = [\varphi \ \vartheta \ \psi]^T$ . La localización (posición y orientación) de nuestro robot, pues, ya está definida. Nombraremos a los giros u orientaciones como *roll* ( $\varphi$ ), *pitch* ( $\vartheta$ ) y *yaw* ( $\psi$ ). Se pueden ver representados conjuntamente en las Figuras 4 y 5.

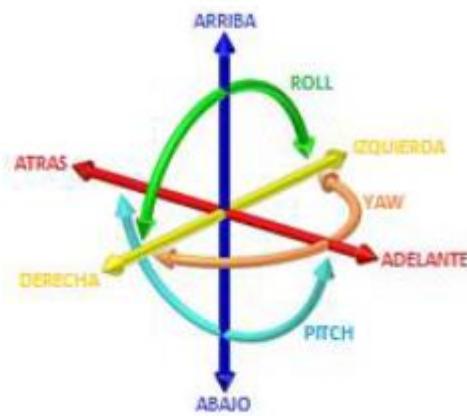


Figura 4. Roll, pitch y yaw. Tomada de [8].

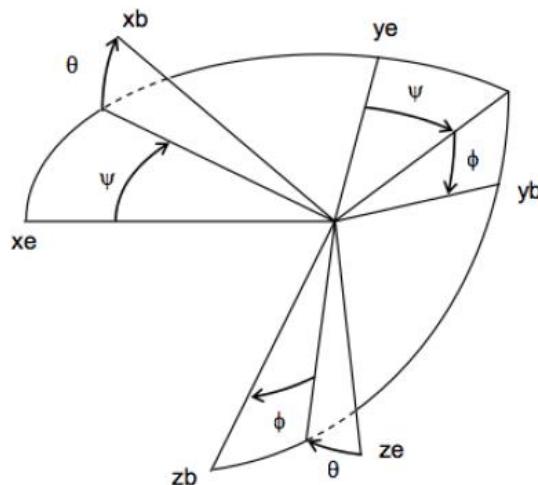


Figura 5. Giro de los ejes. Tomada de [7].

Vayamos con el movimiento. Las velocidades lineales locales se definen como  $u$ ,  $v$  y  $w$ , coherentemente con los ejes X, Y y Z locales, y las velocidades angulares locales, como  $P$ ,  $Q$  y  $R$ , las cuales siguen también la regla de la mano derecha y son respectivas a los giros  $\phi$ ,  $\vartheta$  y  $\psi$ . Así podemos definir también los siguientes vectores:  $\mathbf{v} = [u \ v \ w]^T$  y  $\mathbf{P} = [P \ Q \ R]^T$ . Aparecen reflejados en la Figura 6:

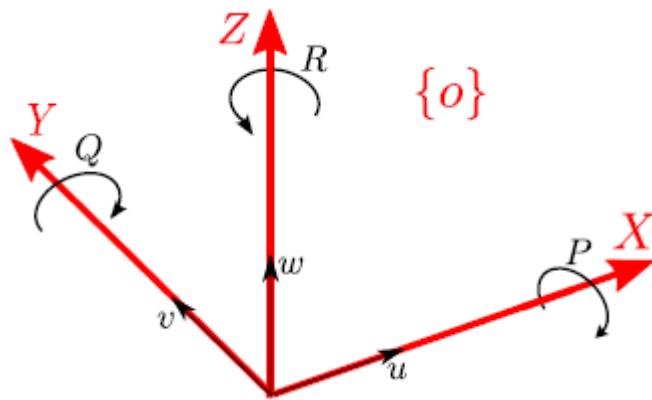


Figura 6. Velocidades lineales y angulares en las coordenadas locales. Tomado de [3].

Podemos relacionar la velocidad en la referencia global,  $\dot{x}$ , con las velocidades en las coordenadas del robot,  $\mathbf{v}$ . Siguiendo la convención de los ángulos de Euler, se define la matriz de rotación de un sistema de referencia a otro como:  $\mathbf{R}(\varphi, \theta, \psi)$ , que es una composición de tres matrices elementales correspondientes a cada uno de los giros:

$$R(X, \varphi) = \begin{pmatrix} 1 & 0 & 0 \\ 0 & c\varphi & -s\varphi \\ 0 & s\varphi & c\varphi \end{pmatrix} \quad (1)$$

$$R(Y, \theta) = \begin{pmatrix} c\theta & 0 & s\theta \\ 0 & 1 & 0 \\ -s\theta & 0 & c\theta \end{pmatrix} \quad (2)$$

$$R(Z, \psi) = \begin{pmatrix} c\psi & -s\psi & 0 \\ s\psi & c\psi & 0 \\ 0 & 0 & 1 \end{pmatrix} \quad (3)$$

Donde se ha definido  $c\psi$  como  $\cos(\psi)$  y  $s\psi$  como  $\sin(\psi)$ . Componiendo las tres matrices obtenemos la matriz de rotación:

$$\begin{aligned} \mathbf{R}(\varphi, \theta, \psi) &= R(Z, \psi) * R(Y, \theta) * R(X, \varphi) \\ &= \begin{pmatrix} c\psi c\theta & c\psi s\theta s\varphi - s\psi c\varphi & c\psi s\theta c\varphi + s\psi s\varphi \\ s\psi c\theta & s\psi s\theta s\varphi + c\psi c\varphi & s\psi s\theta c\varphi - c\psi s\varphi \\ -s\theta & c\theta s\varphi & c\theta c\varphi \end{pmatrix} \end{aligned} \quad (4)$$

Por tanto, podemos obtener ya la primera ecuación, que modela el movimiento lineal:

$$\dot{x} = \mathbf{R}(\varphi, \theta, \psi) * \mathbf{v} \quad (5)$$

Podemos también relacionar las velocidades angulares de la referencia global con las de la referencia local, de la siguiente manera:

$$\dot{\Psi} = \mathbf{R}_A^{-1}(\varphi, \theta, \psi) * \mathbf{P} \quad (6)$$

Siendo  $\mathbf{R}_A^{-1}(\varphi, \theta, \psi)$  la matriz de transformación de referencia para las velocidades angulares:

$$\mathbf{R}_A^{-1}(\varphi, \theta, \psi) = \begin{pmatrix} 1 & t\theta s\varphi & t\theta c\varphi \\ 0 & c\varphi & -s\varphi \\ 0 & \frac{s\varphi}{c\theta} & \frac{c\varphi}{c\theta} \end{pmatrix} \quad (7)$$

### 2.2.2. Modelo dinámico

Como hemos visto, el aumento de la velocidad de uno de los motores, produce un giro en torno a uno de los ejes X o Y, dependiendo de cuál sea éste. Este giro se produce debido a que el empuje, que es una fuerza, de este motor aumenta. Como consecuencia, se produce un par, ya que es una fuerza ubicada fuera del centro de masas del robot. De esta manera, podemos nombrar los pares o torques en torno a los ejes X, Y y Z de la siguiente forma:  $\tau_x$ ,  $\tau_y$  y  $\tau_z$ , respectivamente, que quedan dentro del vector,  $\tau = [\tau_x \ \tau_y \ \tau_z]^T$ . Su sentido positivo coincide con el de P, Q y R. Las otras dos fuerzas principales que actúan sobre el cuadricóptero son el empuje, T, sobre el sentido positivo del eje Z del robot y la gravedad (multiplicada por la masa total del robot), G, sobre el sentido negativo del eje Z del mundo. En la Figura 7 podemos ver todas estas fuerzas y momentos.

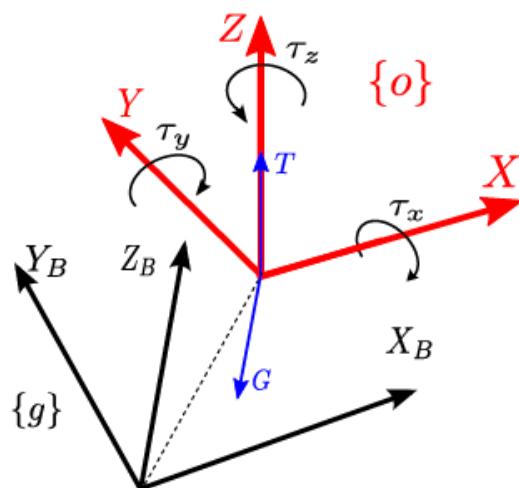


Figura 7. Fuerzas y pares principales. Tomado de [3].

Vamos a definir ahora la fuerza de empuje que ejerce cada una de las hélices. Tal como aparece en [6], la fuerza de empuje depende del cuadrado de la velocidad de cada rotor:

$$F_i = b * \Omega_i^2 \quad (8)$$

donde  $b$  es la constante de empuje con unidades  $\frac{Ns^2}{rad^2}$  y  $\Omega_i$  es la velocidad angular del rotor  $i$ -ésimo, con dimensiones rad/s. El empuje total será, por tanto, la suma de todas estas fuerzas (las de los cuatro motores), tal como se observa en la Figura 8:

$$T = F_1 + F_2 + F_3 + F_4 \quad (9)$$

o escrito de otra manera, combinando las ecuaciones (8) y (9):

$$T = b * (\Omega_1^2 + \Omega_2^2 + \Omega_3^2 + \Omega_4^2) \quad (10)$$

Por tanto, se puede deducir de la Figura 8, con el sentido positivo del eje X alineado con el rotor 2 y el sentido positivo del eje Y alineado con el rotor 1, que el momento alrededor del eje X será producido por el empuje de los motores 1 y 3, y que el momento alrededor del eje Y será producido por el empuje de los motores 2 y 4, de la siguiente forma:

$$\tau_x = l_a * (F_1 - F_3) \quad (11)$$

$$\tau_y = l_a * (F_4 - F_2) \quad (12)$$

Donde  $l_a$  es la longitud de cada brazo del robot, es decir, de cada motor al centro de gravedad.

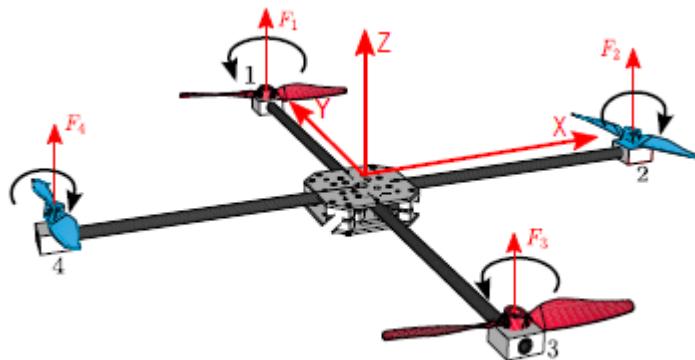


Figura 8. Fuerzas de empuje producidas por los motores. Tomado de [3].

Como hemos explicado anteriormente de manera cualitativa, existe un tercer momento producido por la conservación del momento angular, alrededor del eje Z, que

depende también del cuadrado de la velocidad de cada rotor, de acuerdo con lo que expone [6]:

$$M_i = d * \Omega_i^2 \quad (13)$$

donde  $d$  es la constante de arrastre con unidades de  $\frac{Nms^2}{rad^2}$ . Dado que los rotores giran en sentido contrario dos a dos, el par total definido en el sentido positivo del eje Z, será:

$$\tau_z = -M_1 + M_2 - M_3 + M_4 \quad (14)$$

o bien, combinando las ecuaciones (13) y (14):

$$\tau_z = d * (-\Omega_1^2 + \Omega_2^2 - \Omega_3^2 + \Omega_4^2) \quad (15)$$

Tomando lo anterior podemos agruparlas en forma matricial de la siguiente manera, combinando las ecuaciones (10), (11), (12) y (15):

$$\begin{bmatrix} \tau_z \\ \tau_x \\ \tau_y \end{bmatrix}^T = A * \Omega_S \quad (16)$$

Donde A es la matriz de actuación

$$A = \begin{bmatrix} b & b & b & b \\ bl_a & 0 & -bl_a & 0 \\ 0 & -bl_a & 0 & bl_a \\ -d & d & -d & d \end{bmatrix} \quad (17)$$

Y  $\Omega_S$  es el vector de velocidades angulares cuadradas:

$$\Omega_S = [\Omega_1^2 \ \Omega_2^2 \ \Omega_3^2 \ \Omega_4^2]^T \quad (18)$$

Ahora que ya tenemos las fuerzas y momentos principales (dependientes de las velocidades angulares de los rotores), es hora de relacionarlos con el movimiento del robot. Para ello necesitamos las masas e inercias del sistema.

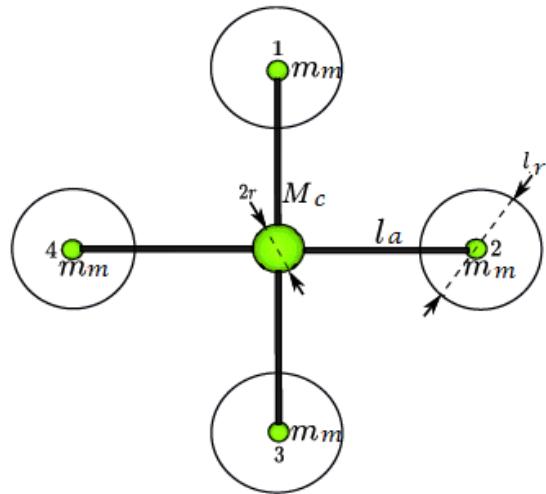


Figura 9. Masas y dimensiones del cuadricóptero. Adaptada de [3].

Tal como vemos en la Figura 9, la masa del centro del robot es  $M_c$ , las masas de los motores (rotores incluidos)  $m_m$ , la distancia de rotor al centro es  $l_a$ , el radio de la placa central (aproximada a una esfera) es  $r$  y el radio del giro del rotor  $l_r$ .

El momento de inercia del robot en el eje X será el mismo que en torno al eje Y, ya que el cuadricóptero es simétrico con respecto a X e Y.

$$I_{xx} = I_{yy} = \frac{2M_c r^2}{5} + 2m_m l_r^2 \quad (19)$$

El momento de inercia respecto del eje Z es diferente:

$$I_{zz} = \frac{2M_c r^2}{5} + 4m_m l_r^2 \quad (20)$$

Con estos tres valores podemos construir la matriz de inercia,  $\mathbf{J}$ , de la siguiente manera, donde sólo se tienen en cuenta los ejes principales de inercia:

$$\mathbf{J} = \begin{bmatrix} I_{xx} & 0 & 0 \\ 0 & I_{yy} & 0 \\ 0 & 0 & I_{zz} \end{bmatrix} \quad (21)$$

Por último, hay que considerar el efecto giroscópico para tener todas las fuerzas que actúan sobre nuestro sistema. Este término depende de las velocidades de las hélices de la siguiente manera:

$$W = I_r * (-\Omega_1 + \Omega_2 - \Omega_3 + \Omega_4) \quad (22)$$

Donde  $I_r$  es la inercia de cada rotor respecto del eje Z, que depende de la masa del rotor  $m_r$  y de la longitud del rotor, o su radio de giro,  $l_r$ :

$$I_r = \frac{m_r l_r^2}{12} \quad (23)$$

Como hemos dicho, necesitábamos las masas e inercias para relacionar las fuerzas y momentos con el movimiento del robot. Ahora ya podemos proceder, tal como aparece en [3], con el movimiento lineal, donde las aceleraciones en la referencia del robot dependen de la fuerza de empuje en el eje Z, de la gravedad (convenientemente transformada a la referencia del robot) y de las velocidades lineales por las velocidades angulares (giros que cambian el módulo y la dirección de la aceleración lineal):

$$\dot{\boldsymbol{v}} = \begin{bmatrix} 0 \\ 0 \\ T \\ M_t \end{bmatrix} + g * \begin{bmatrix} s\theta \\ -s\varphi c\theta \\ -c\varphi c\theta \end{bmatrix} - \mathbf{S}\boldsymbol{v} \quad (24)$$

Donde  $M_t$  es la masa total, suma de la masa de la placa central y las masas de los motores, y  $\mathbf{S}$  es una matriz auxiliar compuesta por velocidades angulares para facilitar la visualización:

$$\mathbf{S} = \begin{bmatrix} 0 & -R & Q \\ R & 0 & -P \\ -Q & P & 0 \end{bmatrix} \quad (25)$$

Seguimos con el movimiento angular, donde la acción de los pares, una vez sustraído el término giroscópico y la fuerza centrípeta, determina la aceleración angular del robot mediante la inercia:

$$\dot{\boldsymbol{P}} = \mathbf{J}^{-1} * \left( \boldsymbol{\tau} - \mathbf{S}\mathbf{J}\boldsymbol{P} - \mathbf{S} \begin{bmatrix} 0 \\ 0 \\ W \end{bmatrix} \right) \quad (26)$$

### 2.2.3. Modelo de los motores

Aunque la mayoría de fuentes no incluye un modelado de los motores que dan acción a las hélices del cuadricóptero, aquí lo haremos, dado que es imprescindible modelar y controlar estos motores de corriente continua (CC), porque las entradas del sistema son las tensiones que se aplican a estos motores.

Nos hemos basado de nuevo en [3] para realizar este modelado. No obstante, es un modelo sencillo, ya que no incluye la inductancia en el motor CC.

El funcionamiento de un motor CC está ampliamente estudiado y consiste en que el par generado por la corriente del circuito, dependiente de la tensión aplicada (a la que se le sustrae la caída de tensión encargada de vencer la fuerza contraelectromotriz), es empleado en mover la inercia. Además, en este caso existe una carga aerodinámica que hay que vencer, y que depende del cuadrado de la velocidad angular.

$$\frac{K_m}{R} * v_i = I_r * \dot{\Omega}_i + \frac{K_m K_e}{R} * \Omega_i + \tau_i \quad (27)$$

Donde  $K_m$  es la constante mecánica del motor, con unidades de  $\frac{Nm}{A}$ ,  $K_e$  es la constante eléctrica del motor, con unidades de  $\frac{Vs}{rad}$ ,  $R$  es la resistencia, con unidades de  $[\Omega]$ , y  $\tau_i$  es la carga aerodinámica del  $i$ -ésimo rotor, que viene determinada por:

$$\tau_i = d * \Omega_i^2 \quad (28)$$

El último aspecto importante a modelar es la tensión máxima que puede dar la fuente de tensión,  $v_{máx}$ , que hará saturar la entrada del sistema y tendrá que ser tomada en cuenta en el control.

Por tanto, como resumen, tenemos las siguientes ecuaciones principales que modelan el comportamiento del cuadricóptero:

$$\dot{x} = R(\varphi, \theta, \psi) * v$$

$$\dot{\Psi} = R_A^{-1}(\varphi, \theta, \psi) * P$$

$$\dot{v} = \begin{bmatrix} 0 \\ 0 \\ T \\ M_t \end{bmatrix} + g * \begin{bmatrix} s\theta \\ -s\varphi c\theta \\ -c\varphi c\theta \end{bmatrix} - S v$$

$$\dot{P} = J^{-1} * \left( \tau - SJP - S \begin{bmatrix} 0 \\ 0 \\ W \end{bmatrix} \right)$$

$$\frac{K_m}{R} * v_i = I_r * \dot{\Omega}_i + \frac{K_m K_e}{R} * \Omega_i + \tau_i$$

## 2.3. Entradas y salidas

Tal como acabamos de ver, las entradas al sistema son las tensiones a aplicar a cada uno de los cuatro motores de corriente continua que dan movimiento a las hélices del cuadricóptero. Estas entradas tienen la limitación de estar saturadas por una tensión máxima que es capaz de dar la fuente de alimentación. Este aspecto será importante y a tener en cuenta a la hora de realizar el control de velocidad de los rotores. Además, vamos a normalizar la entrada para que en vez de un rango de  $0-V_{\max}$  tenga un rango de 0-100 (%). Para ello, simplemente dividiremos cada entrada entre  $V_{\max}$  y la multiplicaremos por 100.

En cuanto a las salidas del sistema, dado que estamos realizando un modelo sencillo, consideraremos que podemos realimentar todas las variables del cuadricóptero, tales como la posición (y sus derivadas), la orientación (y sus derivadas las velocidades y aceleraciones angulares) y las velocidades reales de cada rotor. No consideraremos en el modelado los sensores ni sus respectivos ruidos. Tampoco hemos considerado la acción de un hipotético aleatorio viento, entendida como perturbación al sistema. Se pueden incluir como futuros pasos a seguir del presente trabajo (ver conclusiones).

## 2.4. Verificación del modelo

Para verificar el modelo hay que realizar una serie de experimentos que nos permitan decidir si se ha modelado correctamente el sistema y si se comporta como esperamos que lo haga.

Primero debemos dar un valor numérico a todos los parámetros del sistema, obtenidos de [3]:

PARÁMETRO	VALOR	UDS	PARÁMETRO	VALOR	UDS
Masa de la placa central, $M_c$	0.8211	Kg	Momento de inercia en el eje X	0.0128	$\text{Kgm}^2$
Masa de cada motor, $m_m$	0.1247	Kg	Momento de inercia en el eje Y	0.0128	$\text{Kgm}^2$
Masa total, $M_t$	1.3199	Kg	Momento de inercia en el eje Z	0.0239	$\text{Kgm}^2$
Masa de cada rotor, $M_r$	0.05161	Kg	Constante de empuje, b	9.9865	$\text{Ns}^2/\text{rad}^2$
Radio de cada rotor, $L_r$	0.1	m	Constante de arrastre, d	1.5978	$\text{Nms}^2/\text{rad}^2$

## Modelado, simulación y control de cuadricópteros

Enrique Cañada Panea

Longitud del brazo, l <sub>a</sub>	0.211	m	Constante eléctrica, Ke	0.01	Vs/rad
Radio de la placa central, r	0.07187	m	Constante mecánica, Km	0.01	Nm/A
Momento de inercia del rotor	4.30083 e-5	Kgm <sup>2</sup>	Resistencia, R	0.1107	Ω
Aceleración de la gravedad, g	9.81	1/s <sup>2</sup>	Tensión máxima, Vmáx	11.1	V
Velocidad angular máxima de los motores, Ωmáx	950.276	1/rad			

Antes de realizar la verificación del modelo, se ha desarrollado el control de velocidad angular de los motores (explicado en la siguiente sección) para poder introducir una entrada de velocidad angular, en lugar de tensión, ya que es más cómodo e intuitivo pensar en velocidades angulares que en tensiones, que no sabemos lo que van a significar. Además, debemos calcular las velocidades angulares necesarias para que el cuadricóptero flote, es decir, para vencer a la gravedad, usando la ecuación (10) y suponiendo todas las velocidades iguales:

$$T = Mt * g = 4 * b * \Omega^2$$

$$1.3199 * 9.81 = 4 * 9.9865 * 10^{-6} * \Omega^2$$

$$\Omega = 569.3356 \text{ rad/s}$$
( 29 )

Ahora sí, procedemos a exponer los experimentos, teniendo en cuenta cómo están definidos los ejes, tal como aparece en la Figura 2.

- Introducimos una referencia de 600 rad/s en todos los rotores, con lo que esperaríamos obtener un desplazamiento cuadrático positivo en Z, con su aceleración constante, y desplazamiento nulo en X e Y, así como ángulos  $\varphi$   $\theta$   $\psi$  nulos.

Modelado, simulación y control de cuadricópteros  
 Enrique Cañada Panea

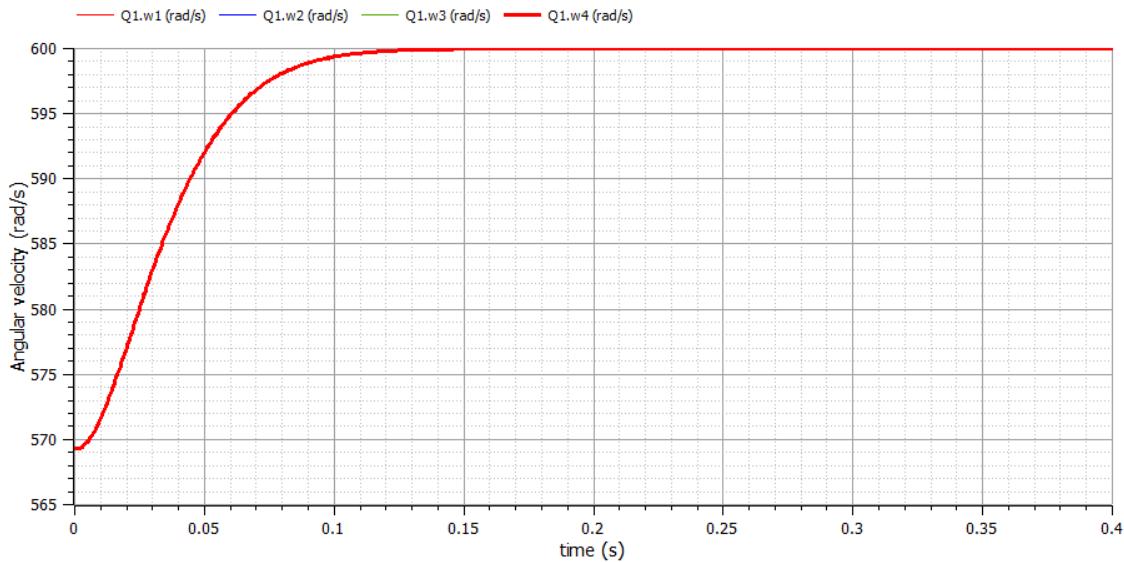


Figura 10. Velocidades angulares de los rotores con referencia de 600 rad/s en todos ellos

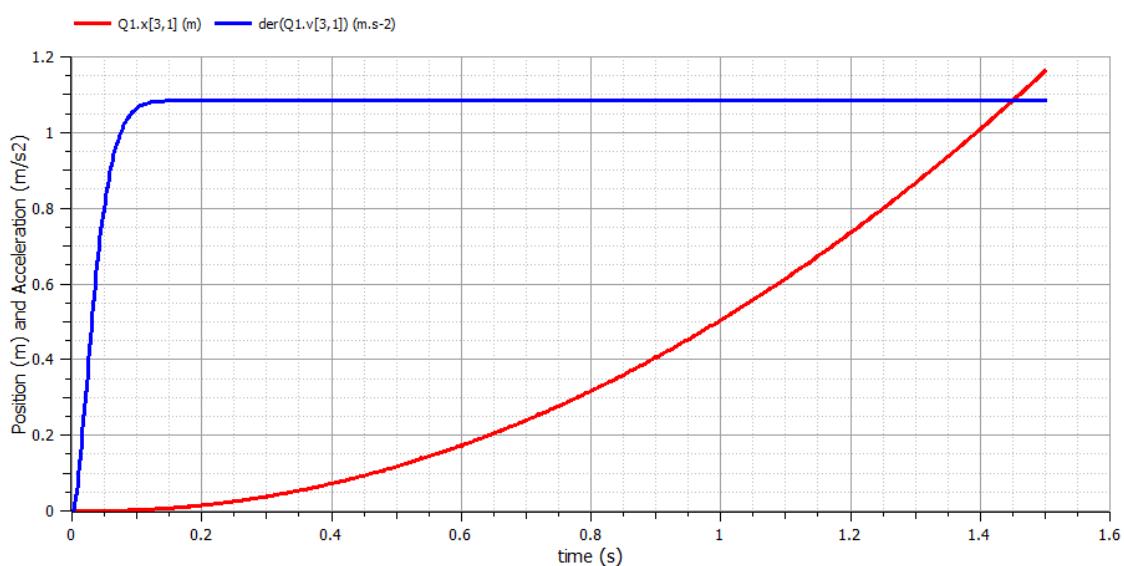


Figura 11. Posición en Z (rojo) y aceleración en Z (azul) con referencia de 600 rad/s en todos los rotores

Como se puede observar en las dos anteriores gráficas, se obtiene el comportamiento deseado anteriormente descrito.

- Referencia de  $\Omega_1 = 700$  rad/s para conseguir un par positivo en el eje X, y por tanto un incremento cuadrático de phi ( $\phi$ ), y una traslación negativa en el eje Y,  $\Omega_3 = 479.78$  rad/s, para contrarrestar el aumento de  $\Omega_1$  y que  $\psi = 0$ . Y las demás como antes:  $\Omega_2 = \Omega_4 = 600$  rad/s.

Modelado, simulación y control de cuadricópteros  
Enrique Cañada Panea

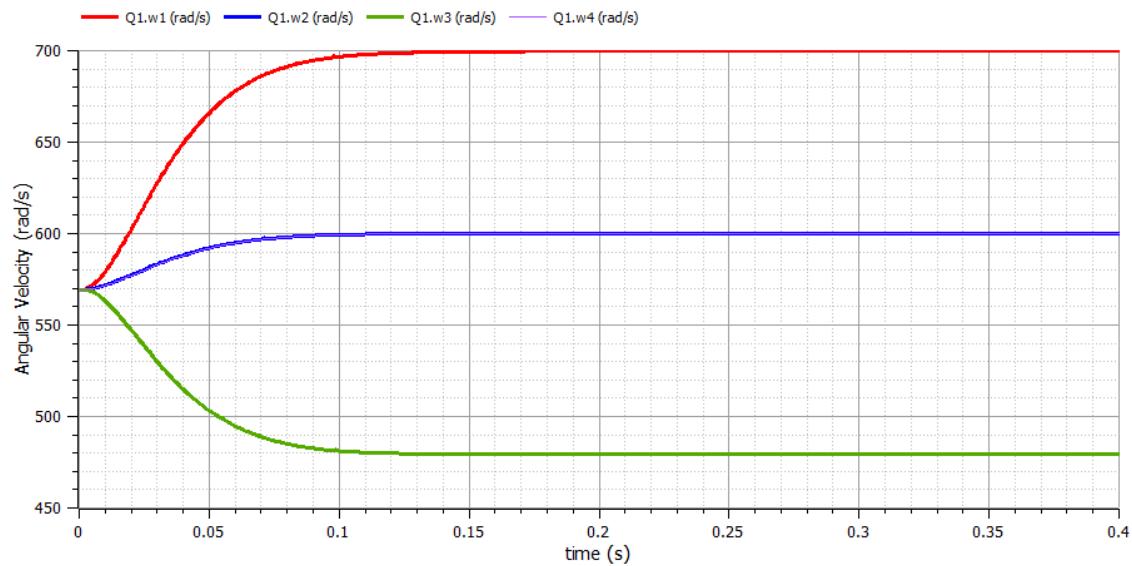


Figura 12. Velocidades angulares de los rotores con referencias de  $\Omega_1 = 700$ ,  $\Omega_2 = \Omega_4 = 600$ ,  $\Omega_3 = 479.78$  rad/s

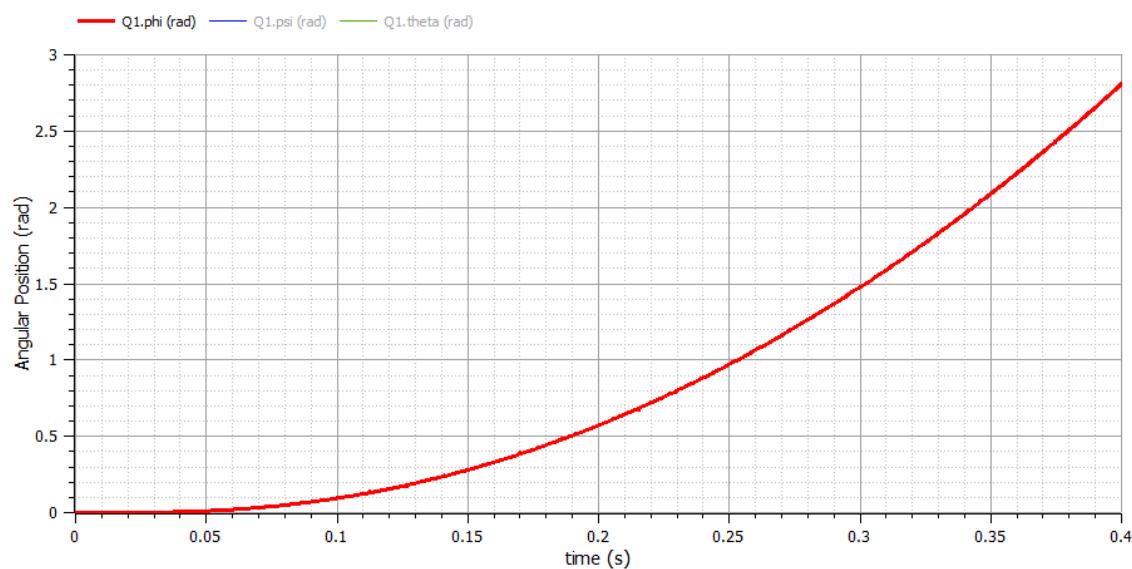


Figura 13. Ángulo  $\phi$  con referencias de  $\Omega_1 = 700$ ,  $\Omega_2 = \Omega_4 = 600$ ,  $\Omega_3 = 479.78$  rad/s

# Modelado, simulación y control de cuadricópteros

## Enrique Cañada Panea

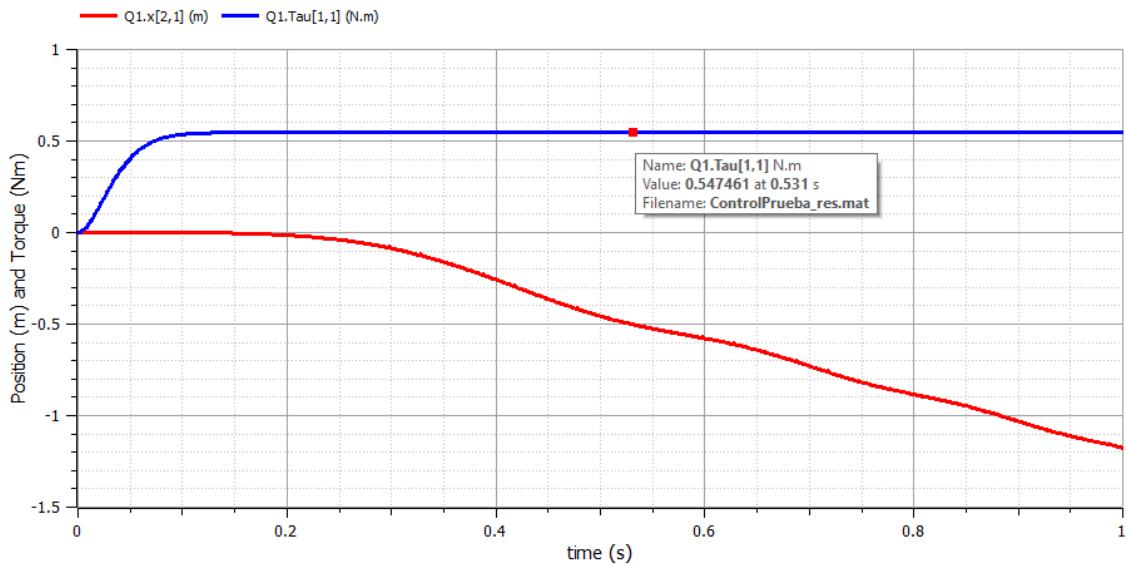


Figura 14. Posición en Y (rojo) y par en el eje X (azul) con referencias de  $\Omega_1 = 700$ ,  $\Omega_2 = \Omega_4 = 600$ ,  $\Omega_3 = 479.78$  rad/s

El resultado vuelve a ser el esperado. El par en el eje X es  $(\Omega_1^2 - \Omega_3^2) * b * l = (700^2 - 479.78^2) * 9.9865 * 10^{-6} * 0.211 = 0.5475$  Nm. Coincide con el valor obtenido.

- Procedimiento inverso para conseguir ahora par negativo alrededor del eje Y y por tanto giro con evolución cuadrática negativa en el tiempo de theta ( $\theta$ ) y una traslación negativa en el eje X:  $\Omega_2 = 800$ ,  $\Omega_4 = 282.84$ ,  $\Omega_1 = \Omega_3 = 600$  rad/s. Tal como antes, están compensadas las velocidades para que no existan otros giros.

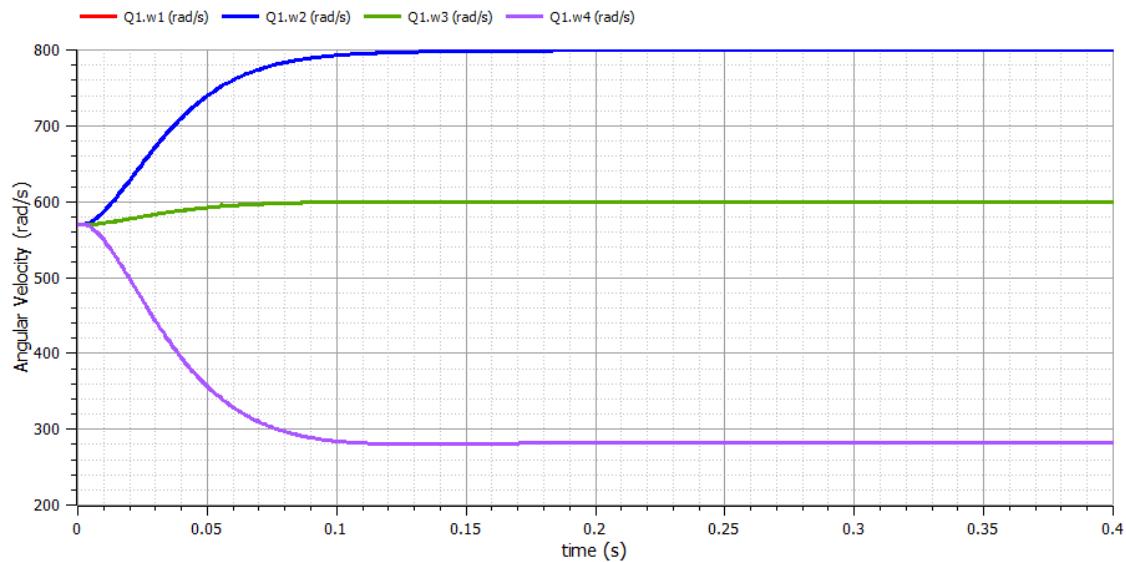


Figura 15. Velocidades angulares de los rotores con referencias de  $\Omega_2 = 800$ ,  $\Omega_1 = \Omega_3 = 600$ ,  $\Omega_4 = 282.84$  rad/s

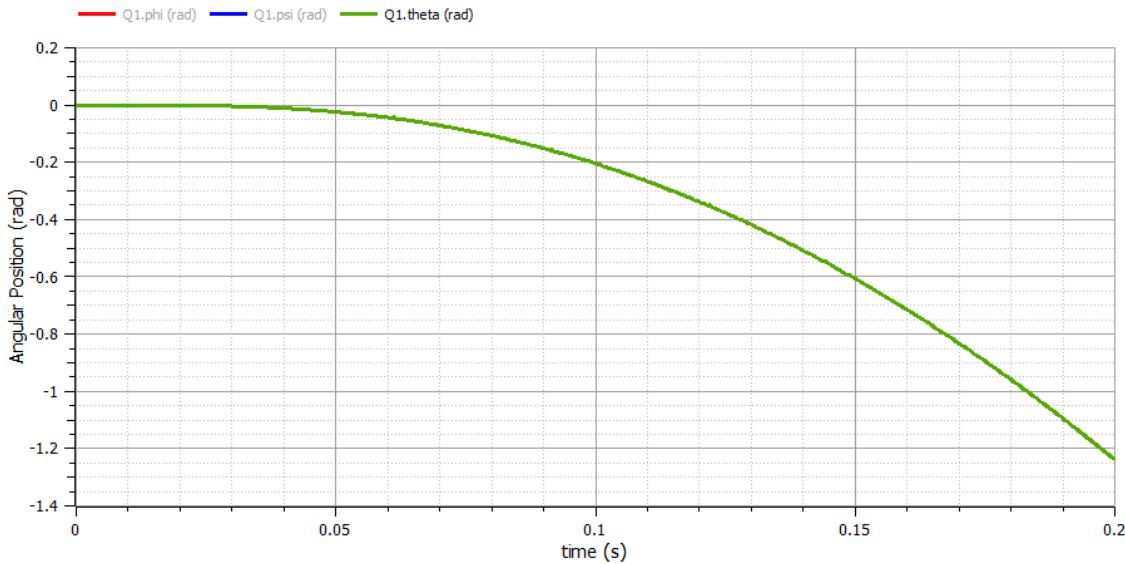


Figura 16. Ángulo theta con referencias de  $\Omega_2 = 800$ ,  $\Omega_1 = \Omega_3 = 600$ ,  $\Omega_4 = 282.84$  rad/s

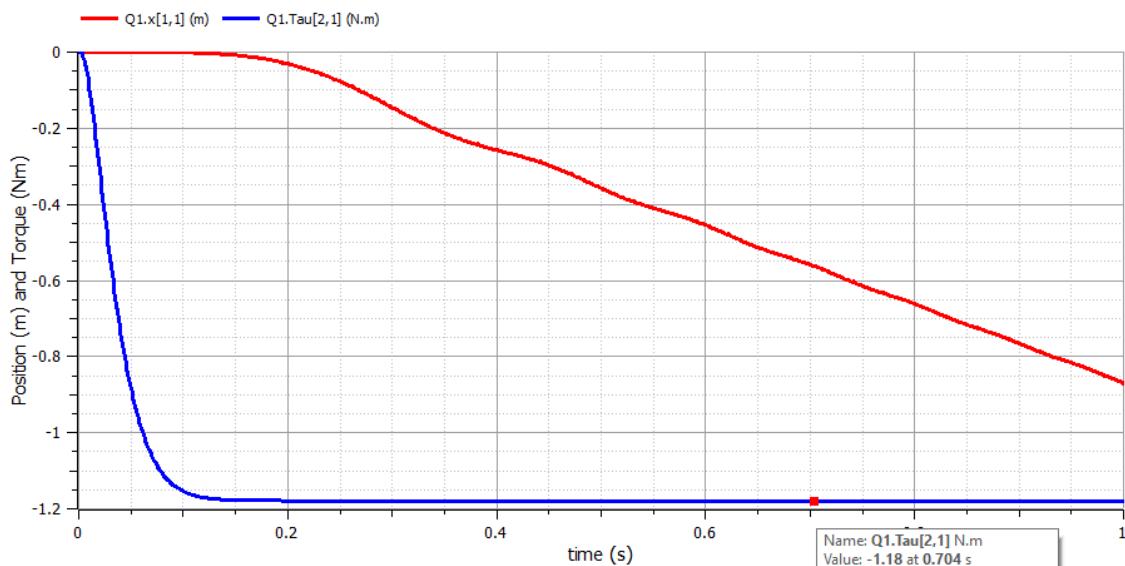


Figura 17. Posición en X (roja) y par en el eje Y (azul) con referencias de  $\Omega_2 = 800$ ,  $\Omega_1 = \Omega_3 = 600$ ,  $\Omega_4 = 282.84$  rad/s

De nuevo obtenemos unos resultados muy similares a lo esperado. De nuevo el valor del par es igual al esperado,  $(\Omega_4^2 - \Omega_2^2) * b * l = (282.84^2 - 800^2) * 9.9865 * 10^{-6} * 0.211 = -1.18$  Nm.

- Procedimiento similar al anterior, pero ahora para producir un par negativo en el eje Z del sistema de referencia del robot, y por tanto obtener un giro negativo cuadrático en psi ( $\psi$ ).  $\Omega_2 = \Omega_4 = 600$ ,  $\Omega_1 = \Omega_3 = 700$  rad/s. Con estas referencias esperamos tener también desplazamiento cuadrático positivo en el eje Z de la referencia global y desplazamiento nulo y giro nulo en los demás ejes.

# Modelado, simulación y control de cuadricópteros

## Enrique Cañada Panea

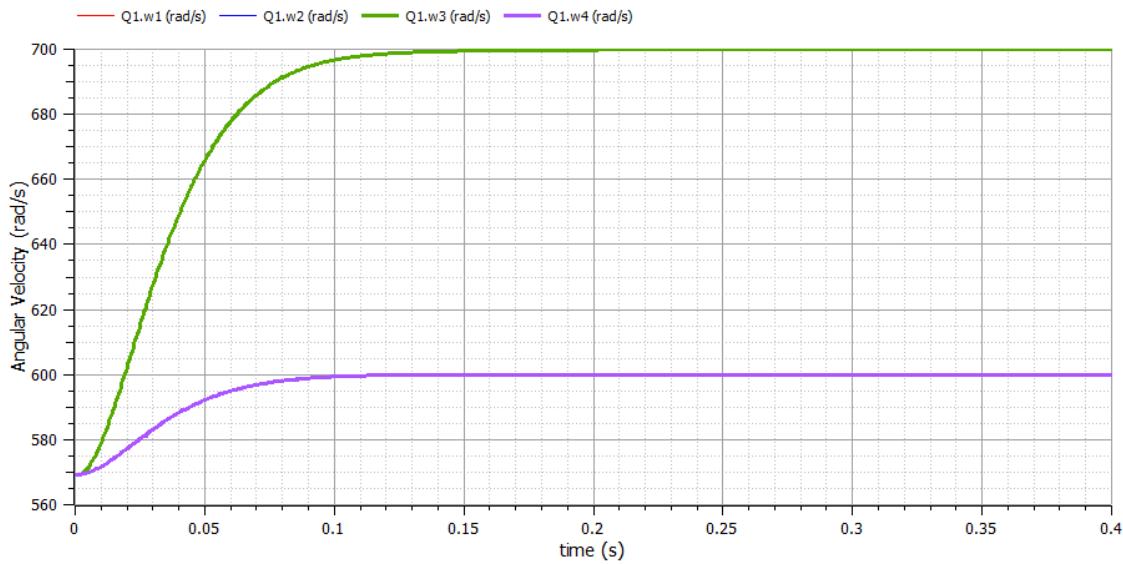


Figura 18. Velocidades angulares de los rotores con referencias de  $\Omega_2 = \Omega_4 = 600$ ,  $\Omega_1 = \Omega_3 = 700$  rad/s

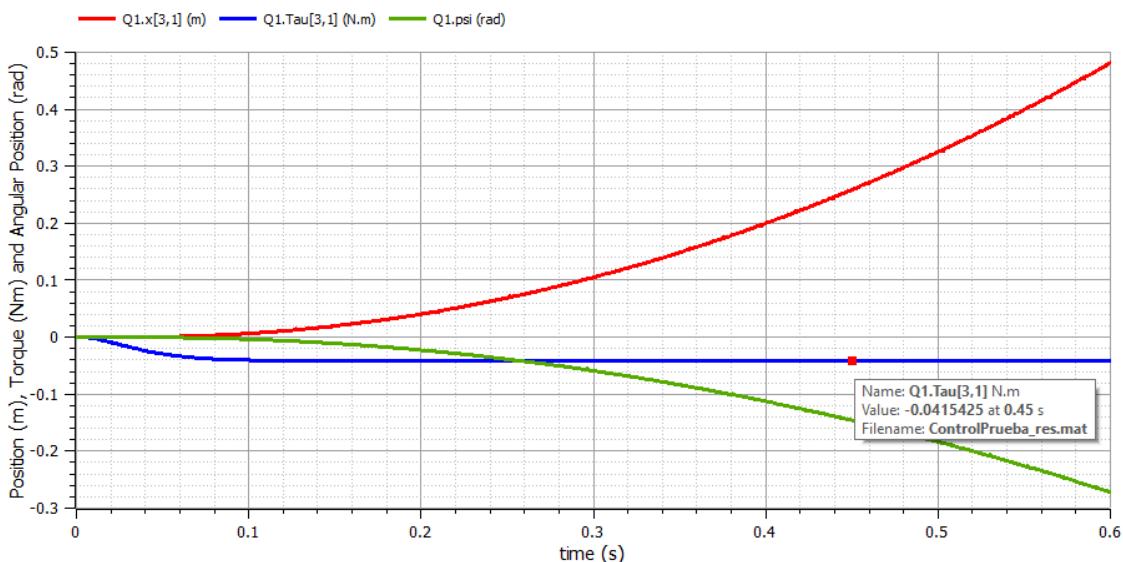


Figura 19. Posición en Z (rojo), par en el eje Z (azul) y ángulo psi (verde) con referencias de  $\Omega_2 = \Omega_4 = 600$ ,  $\Omega_1 = \Omega_3 = 700$  rad/s

Los resultados otra vez reflejan lo que debería pasar con las referencias introducidas. De nuevo se puede comprobar que el par llega al valor esperado:  $(-\Omega_1^2 - \Omega_3^2 + \Omega_2^2 + \Omega_4^2) * d = (-700^2 - 700^2 + 600^2 + 600^2) * 1.5978 * 10^{-7} = -0.0415 \text{ Nm}$ .

Como conclusión de todos los experimentos realizados y de los resultados recogidos, podemos afirmar que el modelo se comporta como debe y como esperábamos que hiciera. Por consiguiente, podemos dar el modelo por verificado.

### 3. Control

El objetivo final de este trabajo consiste en poder controlar las distintas variables del cuadricóptero para llevarlo por una trayectoria espacial deseada. Dicha trayectoria, por supuesto, estará referenciada en el mundo global, ya que son las posiciones de dicha referencia, X, Y y Z, las que un usuario puede contemplar y ver con claridad y le son más familiares.

Para llevar a cabo el deseado gobierno del movimiento del robot, hemos desarrollado varios controles diferentes. Todos ellos son controles diseñados con técnicas clásicas basados en función de transferencia, que incluyen primero en una capa interna un control de velocidad angular de los motores, después un control de ángulos phi, theta y psi (orientación) en una capa exterior y finalmente un control de posición en X, Y y Z como control más externo.

Todos los controles desarrollados son en tiempo discreto y por tanto para calcularlos se han utilizado técnicas de diseño de controladores discretos, o en su defecto han sido discretizados por emulación o no requerían de discretización.

Además, se ha elaborado un generador de trayectorias que desarrolla unos perfiles de posiciones globales ( $x(t)$ ,  $y(t)$  y  $z(t)$ ), de velocidad (las derivadas de las posiciones) y de aceleración (las derivadas de las últimas) suaves y que evitan acciones bruscas que puedan desestabilizar al cuadricóptero o saturar sus entradas como podría pasar con una referencia de escalón, por ejemplo.

Tal como hemos dicho, el control comienza con el control de velocidad de los motores, sigue con el control de orientación y termina con el control de posición. El esquema general quedaría de la siguiente manera (Figura 26).

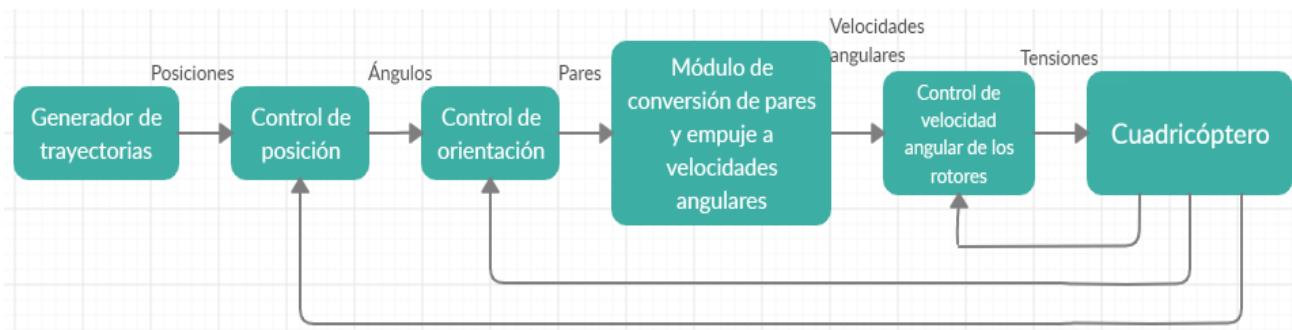


Figura 20. Esquema de control completo

### 3.1. Control de velocidad angular de los rotores

Lo primero de todo es identificar qué se va a controlar y qué ecuaciones rigen su comportamiento. En este caso queremos controlar las velocidades angulares de los rotores, que vienen determinadas únicamente por las entradas de tensión mediante las ecuaciones (27) y (28), aquí en una sola:

$$\frac{K_m}{R} * v_i = I_r * \dot{\Omega}_i + \frac{K_m K_e}{R} * \Omega_i + d * \Omega_i^2 \quad (30)$$

Tenemos un término cuadrático, por lo que deberemos linealizarlo en torno a un punto de equilibrio para poder elaborar la función de trasferencia de esta parte del sistema que consideraremos nuestra planta a controlar para este caso.

Los pasos para linealizar la ecuación son los siguientes:

1. Reordenar la ecuación aislando la variable con derivada:

$$\dot{\Omega} = f(V, \Omega) = \frac{K_m}{R * I_r} * V - \frac{K_m K_e}{R * I_r} * \Omega - d * \Omega^2 \quad (31)$$

2. Elegir un punto de operación. En este caso, la velocidad angular necesaria para que el cuadricóptero se mantenga flotando sin moverse, es decir,  $\bar{\Omega} = 569.3356 \frac{\text{rad}}{\text{s}}$ , dada por la ecuación (29).

3. Calcular el valor de la segunda variable en el punto de operación:

$$\begin{aligned} \dot{\Omega}_i = 0 &= \frac{0.01}{0.1107 * 4.3011 * 10^{-5}} * V \\ &- \frac{0.01 * 0.01}{0.1107 * 4.3011 * 10^{-5}} * 569.3356 \\ &- \frac{1.5978 * 10^{-7}}{4.3008 * 10^{-5}} * 569.3356^2. \end{aligned} \quad (32)$$

$$\bar{V} = 6.2667 \text{ V}$$

4. Hacer el desarrollo en serie de Taylor de la ecuación obviando términos de segundo orden o mayores:

$$\begin{aligned} \dot{\Omega} &\approx \dot{\Omega}(\bar{\Omega}, \bar{V}) + \left. \frac{\partial \dot{\Omega}}{\partial \Omega} \right|_{\Omega=\bar{\Omega}} * (\Omega - \bar{\Omega}) + \left. \frac{\partial \dot{\Omega}}{\partial V} \right|_{V=\bar{V}} * (V - \bar{V})d = \\ &0 - \left( \frac{2d}{I_r} * \bar{\Omega} + \frac{K_m K_e}{R * I_r} \right) * (\Omega - \bar{\Omega}) + \frac{K_m}{R * I_r} * (V - \bar{V}) \end{aligned} \quad (33)$$

5. Sustituir variables, operar y obtener la función de transferencia:

$$\dot{\omega} = -(4.2303 + 21.0039) * \omega + 2100.39 * v \quad (34)$$

$$\frac{\omega}{v} = \frac{2100.39}{s + 25.2342} = \frac{83.236}{0.0396 * s + 1} \quad (35)$$

Otra manera rápida y útil de obtener la función de transferencia del sistema es mediante la identificación. Procedemos a introducir una entrada de escalón de 10 voltios en uno de los motores para observar su respuesta.

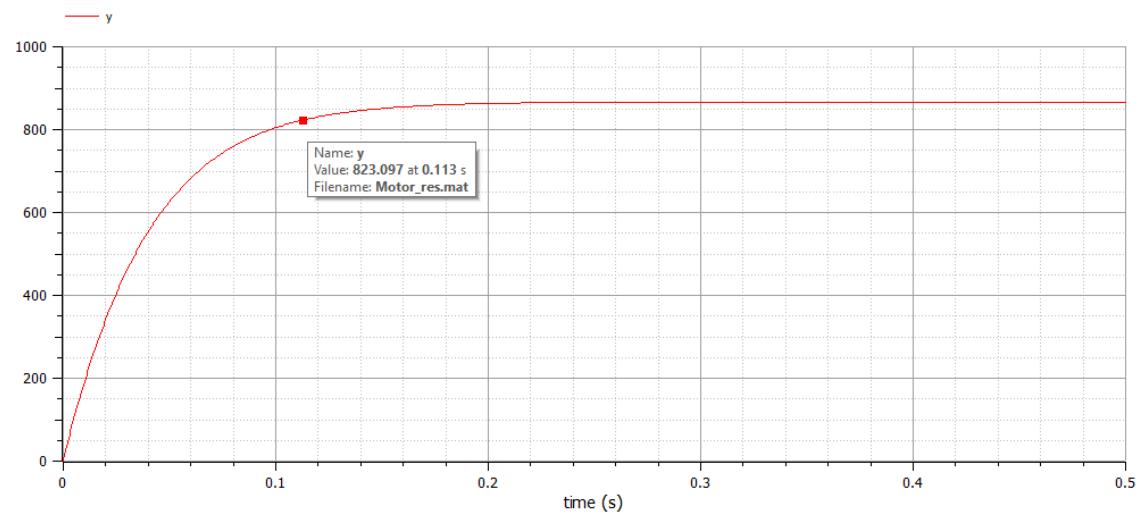


Figura 21. Respuesta al escalón de 10V de un motor.

Lo primero que observamos en la Figura 21 es que tiene un comportamiento de sistema de primer orden. Además, vemos que el valor en régimen permanente es de 866 rad/s y que el tiempo de respuesta (tiempo que tarda la variable en alcanzar el 95% de su valor final) es de 0.113 segundos. Por lo que ya tenemos los valores necesarios para deducir la función de transferencia del sistema.

$$K = 866/10 = 86.6 \text{ frente a } 83.3 \text{ resultado de la linealización}$$

$$T = tr/3 = 0.113/3 = 0.0377 \text{ frente a } 0.0397 \text{ de la linealización, que se puede aproximar a } 0.04.$$

La función de transferencia obtenida mediante este método es:

$$G(s) = \frac{86.6}{0.0377s + 1} = \frac{2297.08}{s + 26.53} \quad (36)$$

Observamos que los resultados son muy similares.

Sin embargo, con el objetivo de normalizar las entradas a nuestro sistema, el cuadricóptero, a un rango 0-100, tal como hemos explicado anteriormente, debemos multiplicar  $G(s)$  por  $V_{\max}/100$ .

$$\begin{aligned} G(s) &= \frac{86.6 * 11.1/100}{0.04s + 1} = \frac{9.6126}{0.04s + 1} \\ &= \frac{240.315}{s + 25} \end{aligned} \quad (37)$$

Ya podemos proceder a la elaboración del controlador, que va a ser llevada a cabo tanto por emulación como por síntesis directa con el control deadbeat.

### 3.1.1. Control diseñado por emulación

Primero realizaremos el cálculo del controlador en continuo y después lo discretizaremos por emulación por el método de emparejamiento de polos y ceros. En este caso el controlador continuo va a ser un compensador. Las especificaciones son:

- Sobreoscilación nula (0%), puesto que al tratarse de un control interno debe ser lo más exacto posible. Por tanto,  $\xi = 1$ .
- Tiempo de respuesta menor de 0.1s, para mejorar la dinámica ya existente (que tiene tiempo de respuesta de 0.113).  
Por tanto  $\sigma > \frac{4.6}{tr} = 46$ . De manera que  $\sigma = \omega_n * \xi$ ;  $\omega_n > 46$ . Elegimos  $\omega_n = 50$ .
- Error de posición nulo. Hay que introducir un integrador en el regulador.
- Tiempo de muestreo inferior a 10 veces el tiempo de subida, es decir,  $T < \frac{1.8}{\omega_n * 10} = 0.0036$ . Escogemos un tiempo de muestreo  $T = 0.003s$ .

Este tiempo de muestreo es posible dentro de la simulación, pero quizá sea demasiado pequeño para una implementación en un controlador real y habría que plantearse aumentarlo en tal caso. El compensador,  $R(s)$ , cancelará el polo del sistema, que es “lento”, mediante un cero. Además introducirá un integrador y un polo lo suficientemente rápido para que  $\omega_n = 50$ . Por consiguiente, la multiplicación de  $R(s)$  y  $G(s)$  tendrá un polo en el origen y otro polo alejado hacia la parte negativa del eje real. Como sabemos que con dos polos el lugar de las raíces forma una cruz, con los polos acercándose conforme aumenta la  $K$  de  $R(s)$  y saliendo después perpendiculares hacia

arriba y hacia abajo, situaremos el polo rápido en  $s = -100$  para que el lugar donde los dos polos se junten sea  $s = -50$  ( $\omega_n = 50$ ,  $\xi = 1$ ).

$$R(s) = \frac{K_R * (s + 25)}{s * (s + 100)} \quad (38)$$

Con MATLAB, realizamos el lugar de las raíces del conjunto  $R(s)*G(s)$ , en la Figura 28:

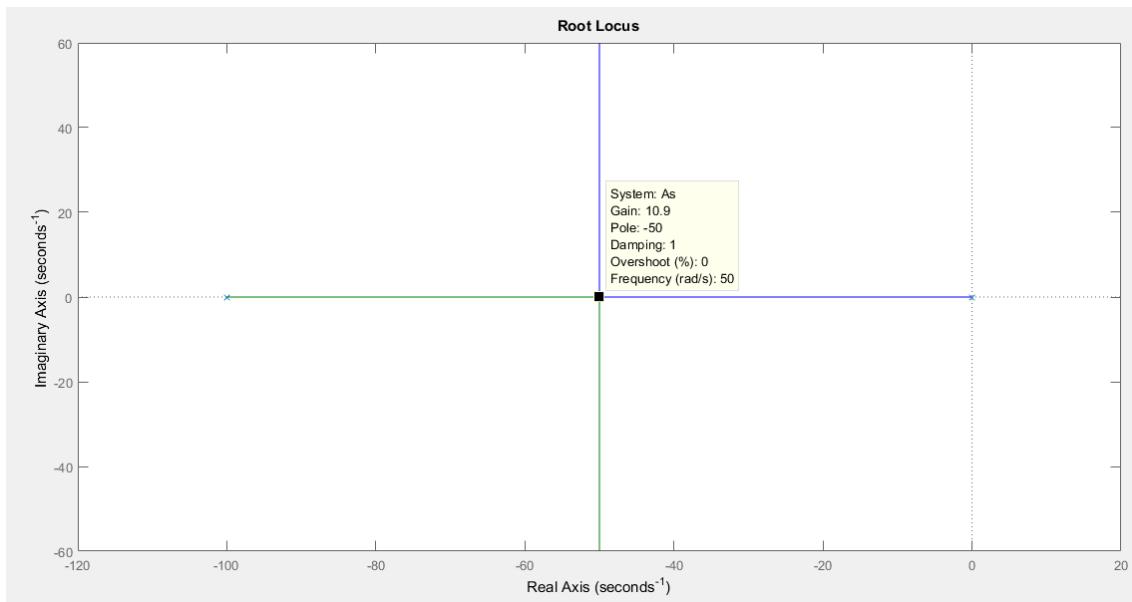


Figura 22. Lugar de las raíces del sistema  $R(s)*G(s)$

Como vemos en el lugar de las raíces, debemos elegir una  $K_r$  de 10.9 para cumplir con las especificaciones deseadas. El regulador queda de la forma:

$$R(s) = \frac{10.9 * (s + 25)}{s * (s + 100)} \quad (39)$$

Ahora que tenemos un buen regulador continuo, vamos a discretizarlo mediante emulación por mapeo de polos y ceros, con el tiempo de muestreo de 0.003s ya determinado anteriormente. Queda el siguiente regulador en z:

$$R(z) = \frac{0.02936z - 0.02724}{z^2 - 1.7408 * z + 0.7408} \quad (40)$$

Vamos a graficar la respuesta al escalón para el bucle cerrado tanto continuo como discreto para compararlos:

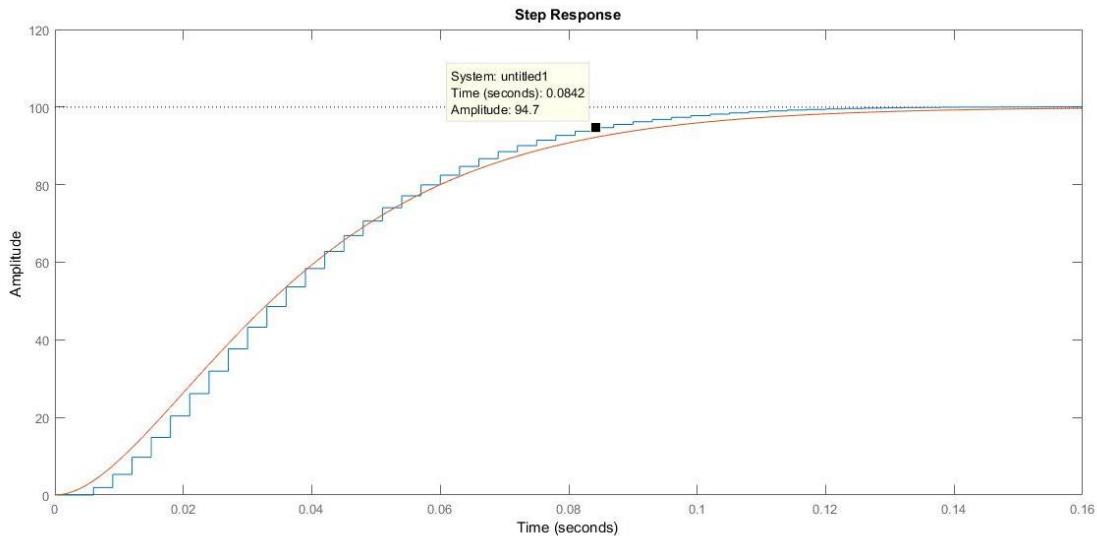


Figura 23. Respuesta al escalón del sistema en bucle cerrado, continuo y discreto, en MATLAB.

Se observa un buen comportamiento de ambos controladores. Incluso el bucle cerrado con el controlador discreto tiene un tiempo de respuesta menor, que es de 0.0842s, tal como se ve en la Figura 23. Se cumplen por tanto las especificaciones: no hay sobreoscilación, el error de posición es nulo, el tiempo de respuesta es sensiblemente menor de 0.1s y el tiempo de muestreo es suficiente para cumplir satisfactoriamente.

Podemos ahora implementarlo mediante un algoritmo en nuestro programa. Si  $R(z) = U(z)/E(z)$ , podemos describir el regulador como una ecuación en diferencias:

$$u(k) = 1.7408 * u(k - 1) - 0.7408 * u(k - 2) + 0.02936 * e(k - 1) - 0.02724 * e(k - 2) \quad (41)$$

Al implementarlo en modélica, se tiene también en cuenta que la entrada satura a 100 como valor máximo y a 0 como valor mínimo. Debido a esta saturación de las entradas al cuadricóptero, se incluye un antiwindup en el controlador:

```

algorithm
when sample(0, T) then
    e_k := e;
    u := 1.7408 * u_k1 - 0.7408 * u_k2 + 0.02936 * e_k1 - 0.02724 * e_k2;
    if u > 100 then
        u := 100;
    end if;
    if u < 0 then
        u := 0;
    end if;
    e_k2 := e_k1;
    e_k1 := e_k;
    u_k2 := u_k1;
    u_k1 := u;
end when;

```

Figura 24. Algoritmo de control de los motores por emulación

Los resultados de la implementación de este controlador en nuestro programa con el modelo completo (sin linealizar), lo veremos más adelante junto con los resultados del control deadbeat (descrito a continuación) para poder compararlos y sacar conclusiones.

### 3.1.2. Control deadbeat

La elaboración del control deadbeat consiste en encontrar qué señal se debe introducir como entrada a un sistema para conseguir llevar la salida al régimen permanente en el mínimo número de periodos. Dicho número de periodos será igual al orden de la planta, para un sistema lineal. Como deseamos una dinámica muy rápida, la más rápida posible, hay que conseguir llevar los polos del sistema en bucle cerrado al origen del plano z. El número de dichos polos será igual a la diferencia entre el orden del denominador y el orden del numerador de la planta discretizada. Procedamos al cálculo del controlador:

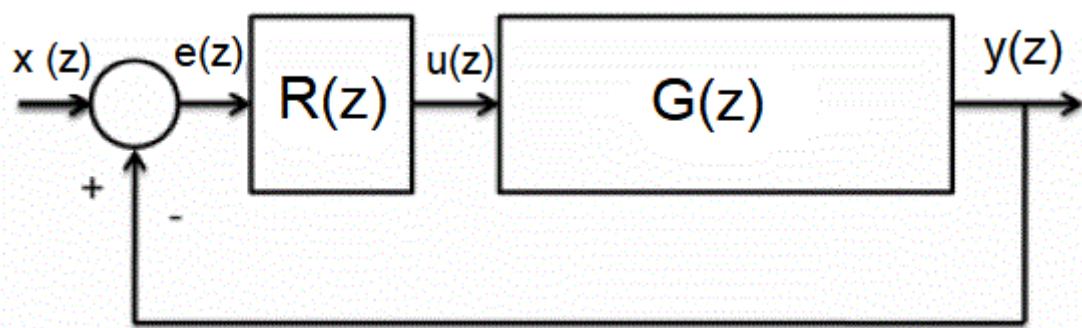


Figura 25. Esquema de control deadbeat. Adaptada de [12]

1. Recordamos la función de transferencia de nuestra planta linealizada:

$$G(s) = \frac{9.6126}{0.0377s + 1} \quad (42)$$

2. Discretizamos la planta:

$$G(z) = \frac{0.7353}{z - 0.9235} \quad (43)$$

3. Como la diferencia entre el orden del denominador y el orden del numerador de la planta es 1, queremos una dinámica del bucle cerrado con un polo en el origen:

$$D(z) = z^{-1} = \frac{R(z) * G(z)}{1 - R(z) * G(z)} \quad (44)$$

4. Calculamos el controlador:

$$\begin{aligned} R(z) &= \frac{1}{G(z)} * \frac{D(z)}{1 - D(z)} = \frac{z - 0.9235}{0.7353} * \frac{z^{-1}}{1 - z^{-1}} = \\ &= \frac{1.36 - 1.25596 * z^{-1}}{1 - z^{-1}} \end{aligned} \quad (45)$$

5. Convertimos a ecuación en diferencias:

$$R(k) = \frac{U(k)}{E(k)} \quad (46)$$

$$U(k) = U(k - 1) + 1.36 * E(k) - 1.25596 * E(k - 1)$$

6. Implementamos en nuestro programa, incluyendo la saturación de las entradas:

```
algorithm
when sample(0, T) then
    e_k := e;
    u := u_k1 + 1.36 * e_k - 1.25596 * e_k1;
    if u > 100 then
        u := 100;
    end if;
    if u < 0 then
        u := 0;
    end if;
    e_k1 := e_k;
    u_k1 := u;
end when;
```

Figura 26. Algoritmo de control deadbeat de los motores

### 3.1.3. Comparación de ambos métodos

Con el fin de comparar ambos métodos de diseño de controladores de los motores anteriormente descritos, vamos a probar cuál es la respuesta en bucle cerrado del sistema, con la planta completa, a diferentes entradas, para cada uno de los dos controladores. Para que estas entradas sean lo más variadas posibles, vamos a introducir un escalón, una rampa que haga saturar la entrada, una sinusode y un tren de pulsos. Graficaremos tanto las velocidades angulares de los motores de nuestro modelo (la planta, el cuadricóptero), como las acciones (en porcentaje) de ambos controladores.

- Respuesta al escalón. Consigna de 800 rad/s (en rosa). Respuesta con control por emulación en rojo y con control deadbeat en verde. Acción del control por emulación en verde claro y la del deadbeat en azul.

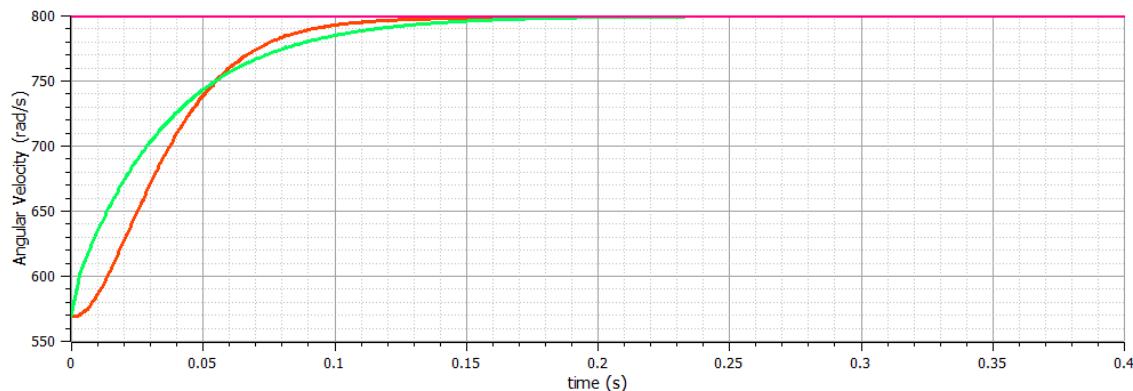


Figura 27. Respuesta al escalón (rosa). Velocidades angulares. Emulación (rojo) y deadbeat (verde)

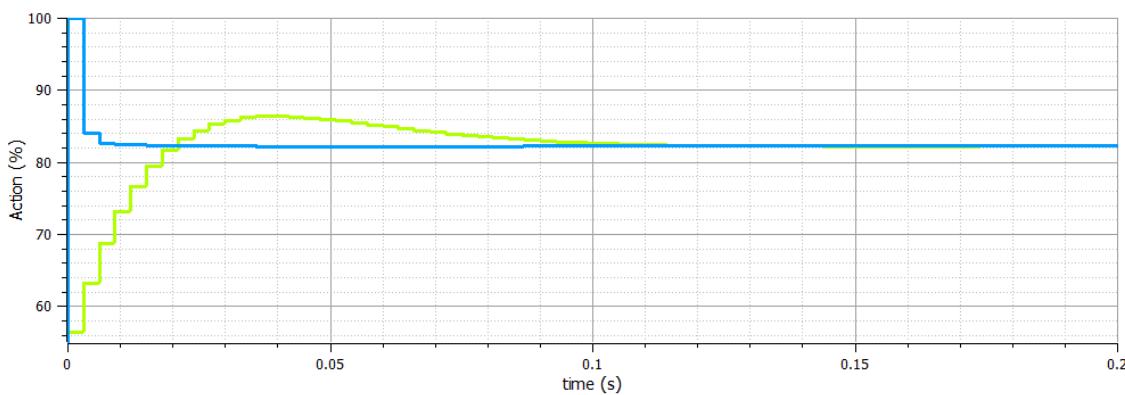


Figura 28. Respuesta al escalón. Acciones. Emulación (verde) y deadbeat (azul)

- Respuesta a una rampa (en amarillo) con saturación de la entrada. Respuesta con control por emulación en rojo y con control deadbeat en verde. Acción del control por emulación en morado y la del deadbeat en azul oscuro.

Modelado, simulación y control de cuadricópteros  
Enrique Cañada Panea

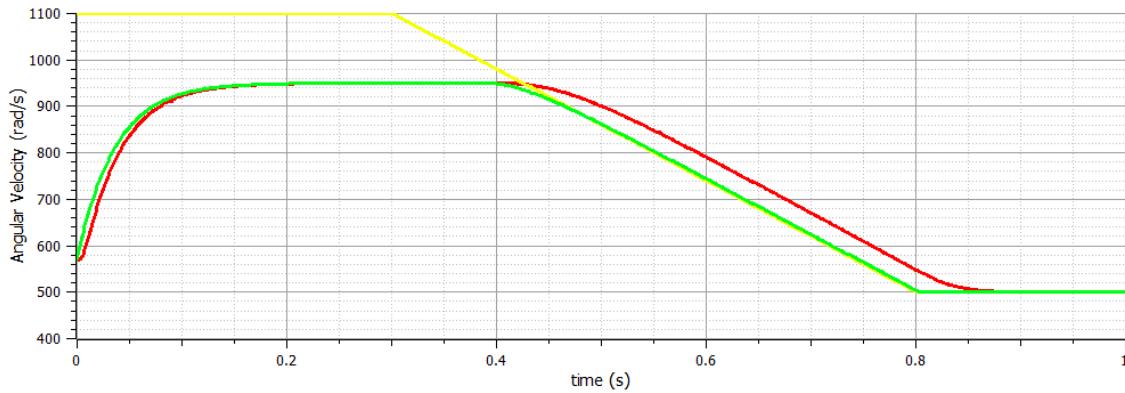


Figura 29. Respuesta a la rampa (amarillo). Velocidades angulares. Emulación (rojo) y deadbeat (verde)

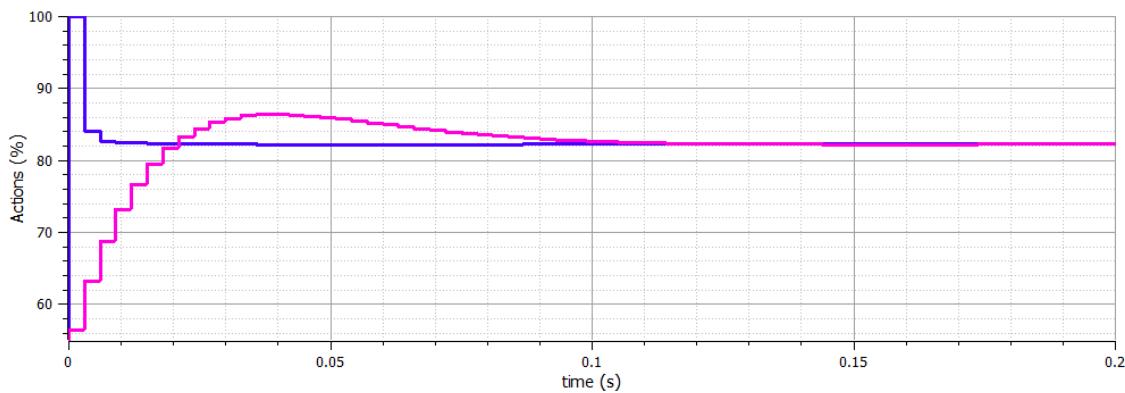


Figura 30. Respuesta a la rampa. Acciones. Emulación (morado) y deadbeat (azul oscuro)

- Respuesta a un tren de pulsos (en morado) de offset 600 rad/s y amplitud 20 rad/s. Respuesta con control por emulación en naranja y con control deadbeat en turquesa. Acción del control por emulación en rosa y la del deadbeat en morado.

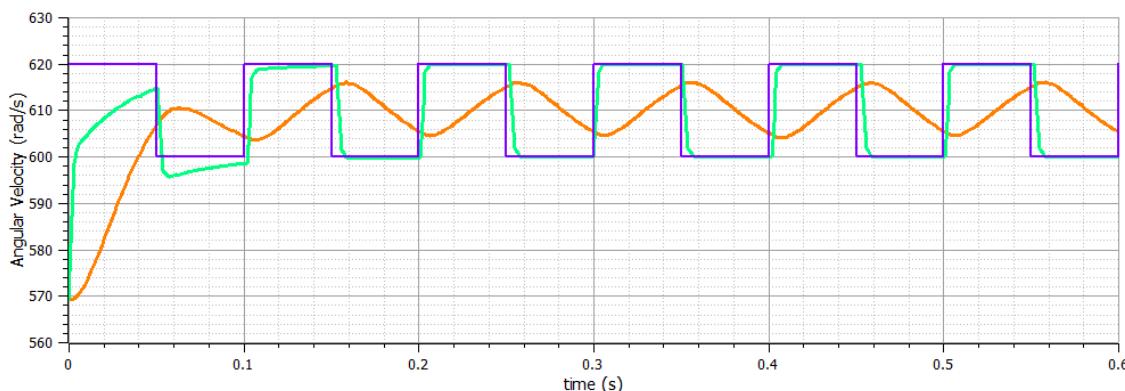


Figura 31. Respuesta a un tren de pulsos (morado). Velocidades angulares. Emulación (naranja) y deadbeat (turquesa)

# Modelado, simulación y control de cuadricópteros

## Enrique Cañada Panea

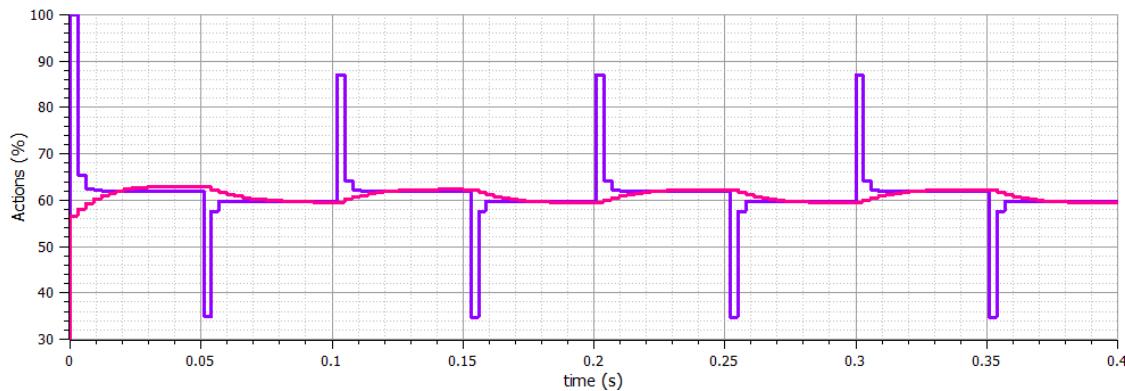


Figura 32. Respuesta a un tren de pulsos. Acciones. Emulación (rosa) y deadbeat (morado)

- Respuesta a una sinusode (en fucsia) de 5 Hz, 600 rad/s de offset y 50 rad/s de amplitud. Respuesta con control por emulación en amarillo y con control deadbeat en azul celeste. Acción del control por emulación en verde y la del deadbeat en azul oscuro.

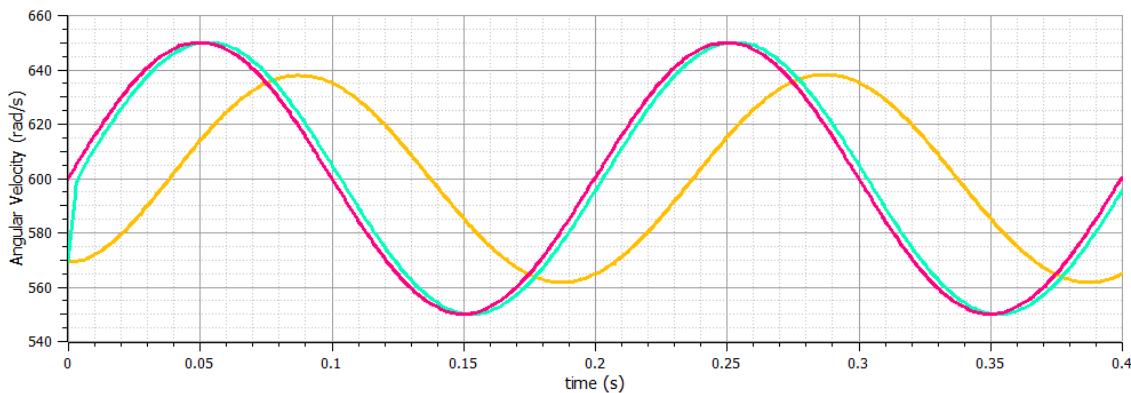


Figura 33. Respuesta a una sinusode (fucsia). Velocidades angulares. Emulación (amarillo) y deadbeat (celeste)

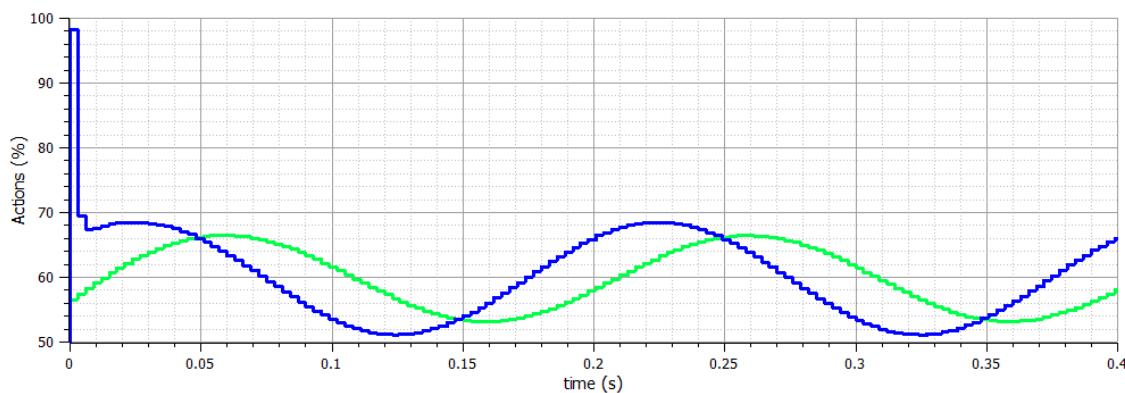


Figura 34. Respuesta a una sinusode. Acciones. Emulación (verde) y deadbeat (azul oscuro)

Tal como vemos en las gráficas anteriores, cuando el cambio de referencia es grande pero no cambia durante un tiempo, como sucede con el escalón, el control

diseñado por emulación llega antes a la referencia (o al 95%), aunque el deadbeat en un principio es más rápido. Sin embargo, cuando la referencia cambia en el tiempo, como los casos de la rampa, la sinusode o el tren de pulsos, vemos que el control deadbeat tiene una respuesta mucho mejor, ya que no tiene prácticamente error. Como la mayor parte del tiempo las referencias de velocidad angular dadas desde aguas arriba del control van a ser cambiantes, se puede concluir que el deadbeat para este caso es un control mejor que el diseñado por emulación y por tanto lo usaremos de aquí en adelante.

### 3.2. Control de orientación

Como en el caso anterior, comenzaremos esta sección enfocando nuestra atención sobre qué hay que controlar y qué ecuación(es) modelan su funcionamiento. Una vez calculado e implementado el control de velocidades de los rotores, pondremos nuestra atención ahora sobre unas variables de un nivel superior, es decir, un poco más cerca del nivel usuario y más lejanas del nivel “máquina”, por hacer una analogía con los lenguajes de programación.

Debemos, pues, controlar la orientación del cuadricóptero, es decir, phi, theta y psi. Ésas serán las referencias que, una vez comparadas con las variables análogas realimentadas, nos darán los errores de posición que constituirán las entradas a nuestro regulador (o reguladores). Las salidas de dicho regulador tenderemos a pensar que son las mismas variables que las entradas al siguiente nivel de control, es decir, las velocidades angulares de las hélices. Sin embargo, facilitará más el control añadir un módulo entre dichos niveles de reguladores, que convierta el empuje  $T$  y los pares  $\tau_x$ ,  $\tau_y$  y  $\tau_z$  en velocidades angulares de rotores, mediante la matriz  $\mathbf{A}$ , definida en la ecuación (17), tal como aparece en [4]. Dicho módulo tendrá como entradas  $T$ ,  $\tau_x$ ,  $\tau_y$  y  $\tau_z$  y como salidas  $\Omega_1$ ,  $\Omega_2$ ,  $\Omega_3$  y  $\Omega_4$ . Su funcionamiento interno, que es una modificación de la ecuación (17) será el siguiente:

$$\begin{aligned}\Omega_1 &= \sqrt{\frac{T}{4 * b} + \frac{\tau_x}{2 * b * l} - \frac{\tau_z}{4 * d}} \\ \Omega_2 &= \sqrt{\frac{T}{4 * b} - \frac{\tau_y}{2 * b * l} + \frac{\tau_z}{4 * d}} \\ \Omega_3 &= \sqrt{\frac{T}{4 * b} - \frac{\tau_x}{2 * b * l} - \frac{\tau_z}{4 * d}}\end{aligned}\quad (47)$$

$$\Omega_4 = \sqrt{\frac{T}{4 * b} + \frac{\tau_y}{2 * b * l} + \frac{\tau_z}{4 * d}}$$

Como consecuencia, las salidas de nuestro regulador serán los tres pares  $\tau_x$ ,  $\tau_y$  y  $\tau_z$ , de los cuales dependen directamente las posiciones angulares phi, theta y psi, que son las variables a controlar. El control de la fuerza T la dejaremos para el control de posición.

Teniendo claro cuáles son las entradas y las salidas del sistema a controlar (las contrarias que las del regulador: pares como entrada y ángulos como salida), ya podemos hacer un pequeño modelo aproximado de la planta que nos ocupa.

Intuitivamente podríamos decir que los pares están relacionados con las posiciones angulares, siguiendo la 2<sup>a</sup> ley de Newton para el movimiento angular, de esta manera:  $\tau_x(s) = I_{xx} * s^2 * \varphi(s)$ , cogiendo el eje X como genérico y haciendo la transformada de Laplace.

Vayamos aun así a las ecuaciones y comprobémoslo. Partimos de la ecuación (26) y la desarrollamos:

$$\dot{\mathbf{P}} = \mathbf{J}^{-1} * \left( \boldsymbol{\tau} - \mathbf{SJP} - \mathbf{S} \begin{bmatrix} 0 \\ 0 \\ W \end{bmatrix} \right)$$

$$\dot{\mathbf{P}} = \begin{bmatrix} \frac{1}{I_{xx}} & 0 & 0 \\ 0 & \frac{1}{I_{yy}} & 0 \\ 0 & 0 & \frac{1}{I_{zz}} \end{bmatrix} * \left( \begin{bmatrix} \tau_x \\ \tau_y \\ \tau_z \end{bmatrix} - \begin{bmatrix} 0 & -R & Q \\ R & 0 & -P \\ -Q & P & 0 \end{bmatrix} * \begin{bmatrix} I_{xx} & 0 & 0 \\ 0 & I_{yy} & 0 \\ 0 & 0 & I_{zz} \end{bmatrix} * \begin{bmatrix} P \\ Q \\ R \end{bmatrix} \right. \\ \left. - \begin{bmatrix} 0 & -R & Q \\ R & 0 & -P \\ -Q & P & 0 \end{bmatrix} * \begin{bmatrix} 0 \\ 0 \\ W \end{bmatrix} \right) \quad (48)$$

$$\dot{\mathbf{P}} = \frac{1}{I_{xx}} * (\tau_x + QR * (I_{yy} - I_{zz}) - Q * W) \quad (49)$$

$$\dot{Q} = \frac{1}{I_{yy}} * (\tau_y + PR * (I_{xx} - I_{zz}) + P * W) \quad (50)$$

$$\dot{R} = \frac{1}{I_{zz}} * (\tau_z + PQ * (I_{yy} - I_{xx})) = \frac{\tau_z}{I_{zz}} \quad (51)$$

Si consideramos velocidades angulares pequeñas, la multiplicación entre ellas se puede despreciar y podemos simplificar estas ecuaciones hasta llegar a la forma  $\tau_x = I_{xx} * s^2 * \varphi$ , deducida intuitivamente antes.

También podemos partir de las ecuaciones de Euler-Lagrange contenidas en [4] para linealizarlas o aproximarlas para al fin simplificarlas, mostradas en la Figura 35.

The matrix  $C(\eta, \dot{\eta})$  has the form, as shown in [9],

$$C(\eta, \dot{\eta}) = \begin{bmatrix} C_{11} & C_{12} & C_{13} \\ C_{21} & C_{22} & C_{23} \\ C_{31} & C_{32} & C_{33} \end{bmatrix},$$

$$\begin{aligned} C_{11} &= 0 \\ C_{12} &= (I_{yy} - I_{zz})(\dot{\theta}C_\phi S_\phi + \dot{\psi}S_\phi^2 C_\theta) + (I_{zz} - I_{yy})\dot{\psi}C_\phi^2 C_\theta - I_{xx}\dot{\psi}C_\theta \\ C_{13} &= (I_{zz} - I_{yy})\dot{\psi}C_\phi S_\phi C_\theta^2 \\ C_{21} &= (I_{zz} - I_{yy})(\dot{\theta}C_\phi S_\phi + \dot{\psi}S_\phi C_\theta) + (I_{yy} - I_{zz})\dot{\psi}C_\phi^2 C_\theta + I_{xx}\dot{\psi}C_\theta \\ C_{22} &= (I_{zz} - I_{yy})\dot{\phi}C_\phi S_\phi \end{aligned} \quad (19)$$

$$\begin{aligned} C_{23} &= -I_{xx}\dot{\psi}S_\theta C_\theta + I_{yy}\dot{\psi}S_\phi^2 S_\theta C_\theta + I_{zz}\dot{\psi}C_\phi^2 S_\theta C_\theta \\ C_{31} &= (I_{yy} - I_{zz})\dot{\psi}C_\theta^2 S_\phi C_\phi - I_{xx}\dot{\theta}C_\theta \\ C_{32} &= (I_{zz} - I_{yy})(\dot{\theta}C_\phi S_\phi S_\theta + \dot{\phi}S_\phi^2 C_\theta) + (I_{yy} - I_{zz})\dot{\phi}C_\phi^2 C_\theta \\ &\quad + I_{xx}\dot{\psi}S_\theta C_\theta - I_{yy}\dot{\psi}S_\phi^2 S_\theta C_\theta - I_{zz}\dot{\psi}C_\phi^2 S_\theta C_\theta \\ C_{33} &= (I_{yy} - I_{zz})\dot{\phi}C_\phi S_\phi C_\theta^2 - I_{yy}\dot{\theta}S_\phi^2 C_\theta S_\theta - I_{zz}\dot{\theta}C_\phi^2 C_\theta S_\theta + I_{xx}\dot{\theta}C_\theta S_\theta. \end{aligned}$$

Equation (18) leads to the differential equations for the angular accelerations which are equivalent with Equations (11) and (12)

$$\ddot{\eta} = J^{-1} (\tau_B - C(\eta, \dot{\eta}) \dot{\eta}). \quad (20)$$

Figura 35. Ecuación de movimiento angular tomada de [4].

Aproximando  $\sin(\theta) \approx \theta$  y  $\cos(\theta) \approx 1$  (para cualquiera de los tres giros), válido para ángulos pequeños (menores de 20º), y aproximando a 0 la multiplicación de dos ángulos o de un ángulo y su derivada (debido a valores pequeños también), podemos simplificar los términos de los elementos de la matriz C:

- $C_{12} = \dot{\psi} * (I_{zz} - 2 * I_{xx})$

- $C13 = 0$
- $C21 = \dot{\psi} * (-I_{zz} + 2 * I_{xx})$
- $C22 = 0$
- $C23 = 0$
- $C31 = -I_{xx} * \dot{\theta}$
- $C32 = \dot{\phi} * (I_{yy} - I_{zz})$
- $C33 = 0$

De esta forma, la ecuación 20 de la Figura 35 queda de la siguiente manera:

$$\ddot{\psi} = J^{-1} * \begin{bmatrix} \tau_x \\ \tau_y \\ \tau_z \end{bmatrix} - \begin{bmatrix} 0 & \dot{\psi} * (I_{zz} - 2 * I_{xx}) & 0 \\ \dot{\psi} * (-I_{zz} + 2 * I_{xx}) & 0 & 0 \\ -I_{xx} * \dot{\theta} & \dot{\phi} * (I_{yy} - I_{zz}) & 0 \end{bmatrix} * \begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} \quad (52)$$

La multiplicación de las dos matrices finales va a producir términos 0 o términos que incluyen multiplicación de derivadas de ángulos, por lo que al aproximarlas el resultado de la multiplicación de dichas matrices será 0.

Por tanto, hemos demostrado de dos formas (aunque incluyendo linealización y simplificación), que nuestra ecuación inicial era correcta (al menos, aproximadamente):  $\tau_x(s) = I_{xx} * s^2 * \varphi(s)$ , válida para los tres giros, con sus variables análogas.

Tenemos que nuestra planta en este caso es:

$$G(s) = \frac{\tau(s)}{\varphi(s)} = \frac{1}{I * s^2} \quad (53)$$

Entendiendo  $\varphi$  como cualquiera de los tres ángulos e  $I$  (el momento de inercia para cada eje) diferente para el caso del ángulo en torno a el eje Z,  $\psi$ .

Vamos a diseñar dos tipos de control, como en el caso del control de los motores, para después comparar sus resultados y sus comportamientos. Por un lado, un control por realimentación taquimétrica y por otro lado un control deadbeat.

### 3.2.1. Realimentación taquimétrica

El objetivo inicial era implementar un control discreto por emulación de un control continuo original diseñado por método frecuencial, pero finalmente se ha optado por la realimentación taquimétrica debido a que el comportamiento de aquél no era satisfactorio.

Se había diseñado un control proporcional avance de fase (PAF), dado que al tener un doble integrador en la planta no es necesario integrar y se puede tener un mejor transitorio. A partir del diagrama de bode de la planta, cuyo diagrama de fases es siempre de  $-180^\circ$  al tener únicamente dos integradores, se deduce que no se dispone de margen de fase ( $M_f = 0$ ). Con una sobreoscilación deseada del 0%, el margen de fase debe ser de  $76^\circ \approx 80^\circ$ . Sin embargo, esto lleva un  $\alpha = 0.0077$  o la aplicación de 2 PAF en serie de  $40^\circ$  o  $42.5^\circ$  cada uno para tener un margen de fase de  $80^\circ$ - $85^\circ$ . Se puede deducir que, con errores grandes o cambios de consigna bruscos, las acciones aplicadas por el regulador serán demasiado grandes.

Implementado en nuestro programa, un PAF de  $\alpha = 0.0077$  funciona correctamente para el control del ángulo  $\psi$  y, para una de las otras dos posiciones angulares  $\phi$  y  $\theta$ , cuando solo cambia la referencia de dicha posición angular. Pero cuando cambia la referencia en dos de ellas o en las tres, el funcionamiento del control de  $\phi$  y  $\theta$  no es el deseado (sobreoscilaciones desmesuradas y errores de posición), debido a las no linealidades de la planta. Tal como aparece en las ecuaciones (49) y (50), cuando las velocidades angulares no son nulas, para  $\phi$  y  $\theta$  aparecen términos que no se tienen en cuenta en el modelo linealizado.

Por tanto, se ha optado por realizar una realimentación taquimétrica que, aunque no produce una respuesta tan rápida, no tiene error de posición.

El esquema es el siguiente:

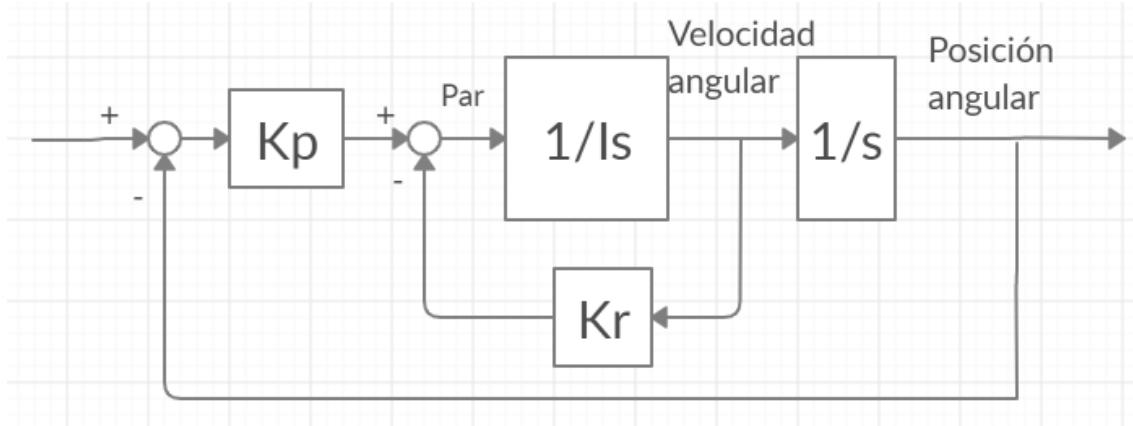


Figura 36. Esquema de control por realimentación taquimétrica

Simplificando este diagrama de bloques, llegamos a la siguiente función de transferencia:

$$F(s) = \frac{\frac{K_p}{I}}{s^2 + \frac{K_R}{I} * s + \frac{K_p}{I}} \quad (54)$$

Podemos deducir que:

$$\omega_n^2 = \frac{K_P}{I} \quad (55)$$

$$2 * \xi * \omega_n = \frac{K_R}{I} \quad (56)$$

Con la especificación de SO = 0% elegimos un  $\xi = 1$  y con la especificación de  $t_r = 1s$ :

$$t_r = \frac{4.75}{\omega_n}; 1 = \frac{4.75}{\omega_n}; \omega_n = 4.75 \text{ rad/s} \quad (57)$$

Podemos ahora introducir los valores de  $\xi$  y  $\omega_n$  en las ecuaciones (55) y (56) para obtener  $K_P = 0.2888$  y  $K_R = 0.1216$  para controlar  $\phi$  y  $\theta$ , y  $K_P = 0.5392$  y  $K_R = 0.227$  para controlar  $\psi$ , dado que tienen distintos momentos de inercia.

Así lo hemos implementado en nuestro programa:

```
parameter Real T = 0.05, Kp = 0.2888, Kd = 0.1216, Kpz = 0.5392, Kdz = 0.227;
algorithm
when sample(0, T) then
    TauPhi := Kp*(Phid - Phi) - Kd*derPhi;
    TauTheta := Kp*(Thetad - Theta) - Kd*derTheta;
    TauPsi := Kpz*(Psid - Psi) - Kdz*derPsi;
end when;
```

Figura 37. Algoritmo de control por realimentación taquimétrica

### 3.2.2. Control deadbeat

De manera similar a como hemos hecho antes con el control de los motores, los pasos para diseñar el control deadbeat de la orientación del cuadricóptero son:

1. Recordamos las funciones de transferencia de nuestras plantas linealizadas:

$$G_x(s) = \frac{\varphi}{\tau_x} = G_y(s) = \frac{\theta}{\tau_y} = \frac{1}{0.0128s^2} \quad (58)$$

$$G_z(s) = \frac{\psi}{\tau_z} = \frac{1}{0.0239s^2} \quad (59)$$

2. Discretizamos las plantas con un periodo de muestreo de 0.05s, suficientemente pequeño para un tiempo de respuesta de 1s:

$$G_x(z) = G_y(z) = \frac{0.097656 * (z + 1)}{(z - 1)^2} \quad (60)$$

$$G_z(z) = \frac{0.0523 * (z + 1)}{(z - 1)^2} \quad (61)$$

3. Como la diferencia entre el orden del denominador y el orden del numerador de la planta discretizada es 1, queremos una dinámica del bucle cerrado con un solo polo en el origen:

$$D(z) = z^{-1} = \frac{R(z) * G(z)}{1 - R(z) * G(z)} \quad (62)$$

4. Calculamos los controladores, para  $\varphi$  o  $\theta$ , y para  $\psi$ , respectivamente:

$$\begin{aligned} R_x(z) &= \frac{1}{G_x(z)} * \frac{D(z)}{1 - D(z)} = \frac{(z - 1)^2}{0.097656 * (z + 1)} * \frac{z^{-1}}{1 - z^{-1}} \\ &= \frac{10.24003 - 10.24003 * z^{-1}}{1 + z^{-1}} \quad (63) \\ R_z(z) &= \frac{(z - 1)^2}{0.0523 * (z + 1)} * \frac{z^{-1}}{1 - z^{-1}} \\ &= \frac{19.1205 - 19.1205 * z^{-1}}{1 + z^{-1}} \end{aligned}$$

5. Convertimos a ecuaciones en diferencias, para  $\varphi$  y  $\theta$ :

$$R(k) = \frac{U(k)}{E(k)} \quad (64)$$

$$U(k) = -U(k - 1) + 10.24003 * (E(k) - E(k - 1))$$

Y para  $\psi$ :

$$U(k) = -U(k - 1) + 19.1205 * (E(k) - E(k - 1)) \quad (65)$$

6. Implementamos en nuestro programa:

```

parameter Real T = 0.05;
Real ux, u_klx(start = 0), e_kx(start = 0), e_klx(start = 0);
Real uy, u_kly(start = 0), e_ky(start = 0), e_kly(start = 0);
Real uz, u_klz(start = 0), e_kz(start = 0), e_klz(start = 0);
algorithm
when sample(0, T) then
//X
  e_kx := Phid - Phi;
  ux := -u_klx + 10.24003 * (e_kx - e_klx);
  e_klx := e_kx;
  u_klx := ux;
  TauPhi := ux;
//Y
  e_ky := Thetad - Theta;
  uy := -u_kly + 10.24003 * (e_ky - e_kly);
  e_kly := e_ky;
  u_kly := uy;
  TauTheta := uy;
//Z
  e_kz := Psid - Psi;
  uz := -u_klz + 19.12046 * (e_kz - e_klz);
  e_klz := e_kz;
  u_klz := uz;
  TauPsi := uz;
end when;

```

---

### 3.2.3. Comparación de ambos métodos

Para realizar la comparación de ambos controles vamos a observar cómo se comporta el sistema con diferentes entradas, como en el caso del control de los motores. Estas entradas serán: una rampa en sólo uno de los tres ángulos y ninguna entrada en los otros dos, una rampa en los tres ángulos a la vez, y mezcla de sinusoide, rampa y tren de pulsos una en cada ángulo.

- Respuesta a la rampa en  $\varphi$ . Referencia en naranja, respuesta con el control deadbeat en verde y respuesta con la realimentación taquimétrica en amarillo.

Modelado, simulación y control de cuadricópteros  
Enrique Cañada Panea

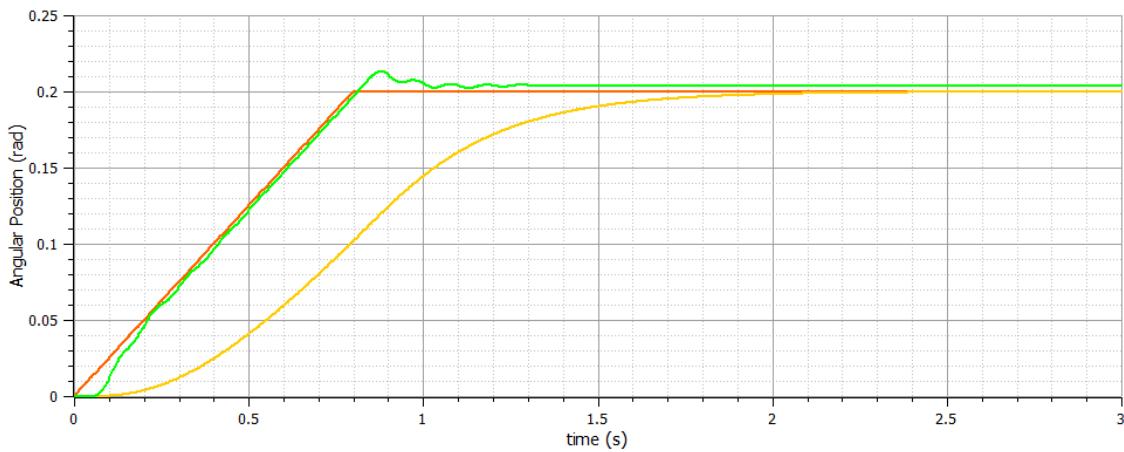


Figura 38. Referencia de rampa en  $\phi$  (naranja). Respuesta deadbeat (verde) y realimentación taquimétrica (amarillo)

Y los pares (acciones):

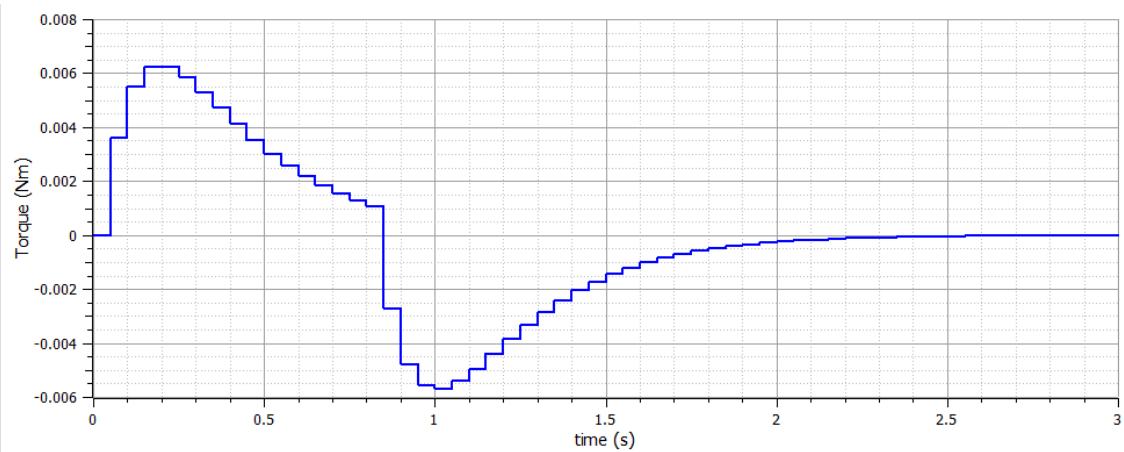


Figura 39. Par ejercido en el eje X con realimentación taquimétrica

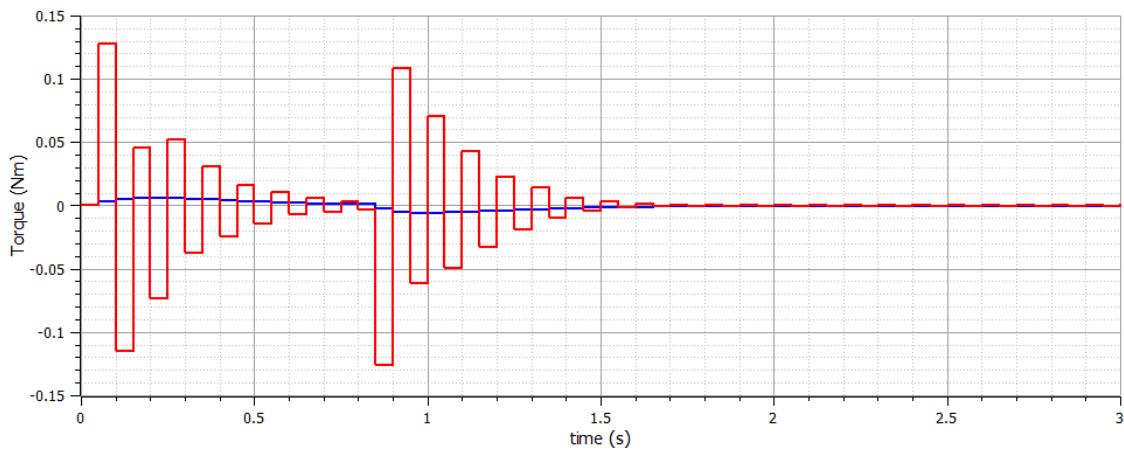


Figura 40. Pares en el eje X con deadbeat (rojo) y realimentación taquimétrica (azul)

- Respuesta a la rampa en  $\theta$ . Referencia en fucsia, respuesta con el control deadbeat en amarillo y respuesta con la realimentación taquimétrica en verde. Los pares son muy similares al caso anterior y no es necesario graficarlos otra vez.

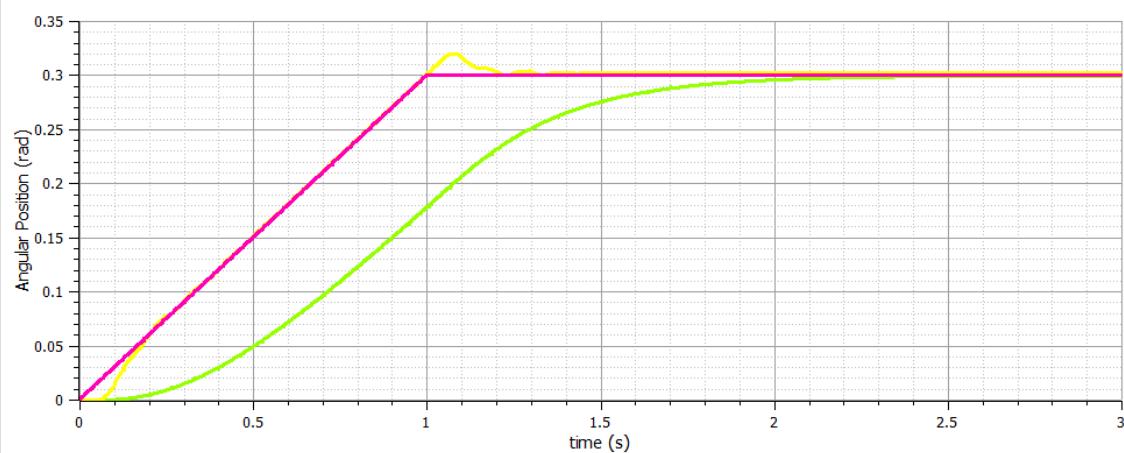


Figura 41. Referencia de rampa en theta (fucsia). Respuesta deadbeat (amarillo) y realimentación taquimétrica (verde)

- Respuesta a la rampa en  $\psi$ . Referencia en rojo, respuesta con el control deadbeat en verde y respuesta con la realimentación taquimétrica en amarillo.

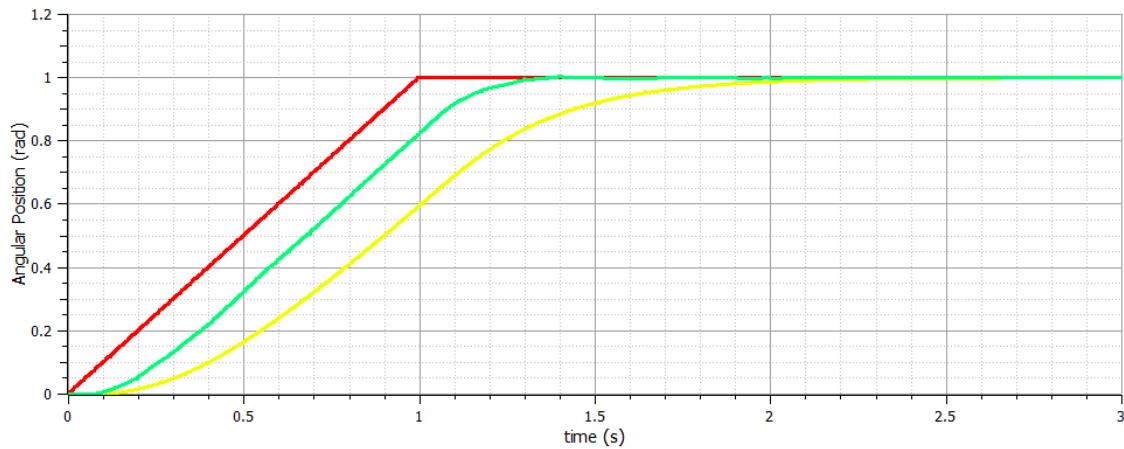


Figura 42. Referencia de rampa en  $\psi$  (rojo). Respuesta deadbeat (verde) y realimentación taquimétrica (amarillo)

Modelado, simulación y control de cuadricópteros  
 Enrique Cañada Panea

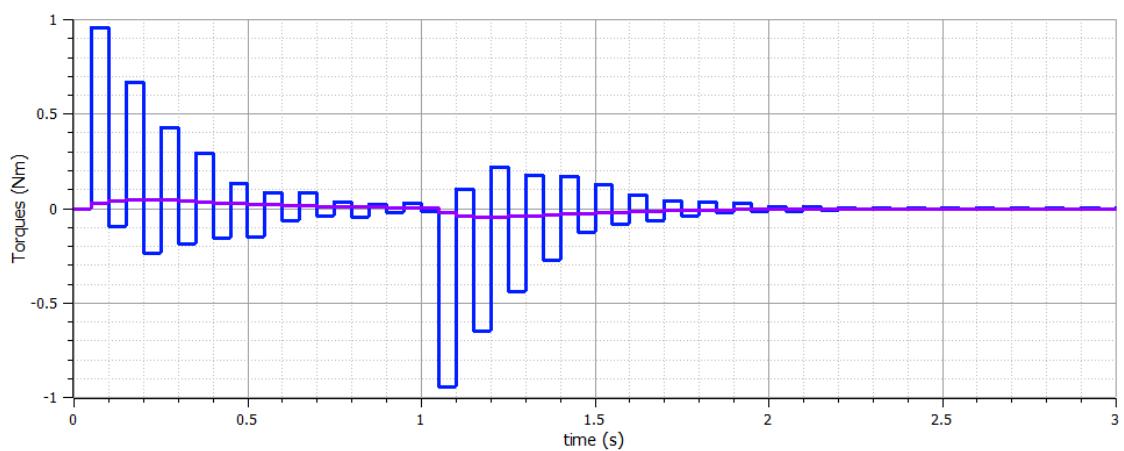


Figura 43. Pares en el eje Z con deadbeat (azul) y realimentación taquimétrica (morado)

- Respuesta a tres rampas en  $\varphi$ ,  $\theta$  y  $\psi$ . Referencia de  $\theta$  en negro, respuesta con el control deadbeat en rojo y respuesta con la realimentación taquimétrica en azul oscuro. Referencia de  $\varphi$  en gris, respuesta con el control deadbeat en naranja y respuesta con la realimentación taquimétrica en azul claro.

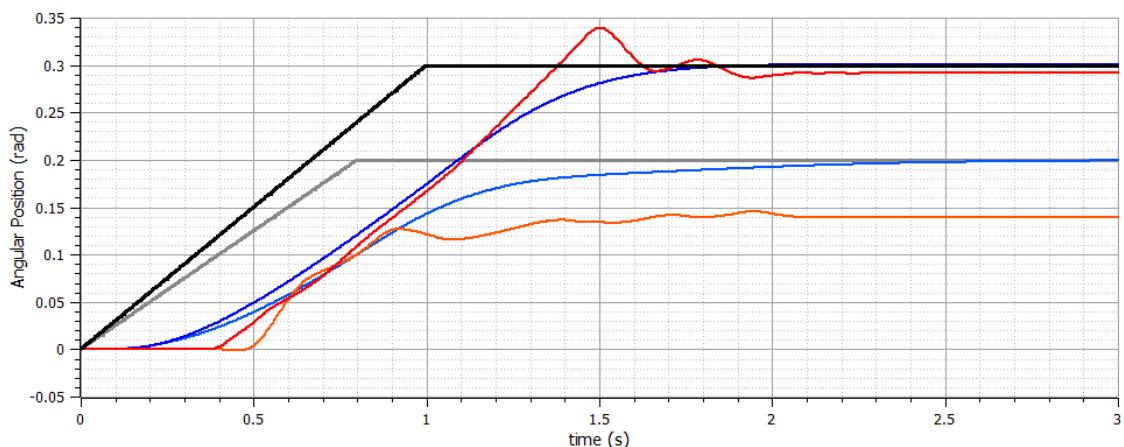


Figura 44. Respuesta a rampas en los tres ángulos.  $\Phi$  y  $\theta$

Referencia de  $\psi$  en rojo, respuesta con el control deadbeat en verde y respuesta con la realimentación taquimétrica en amarillo.

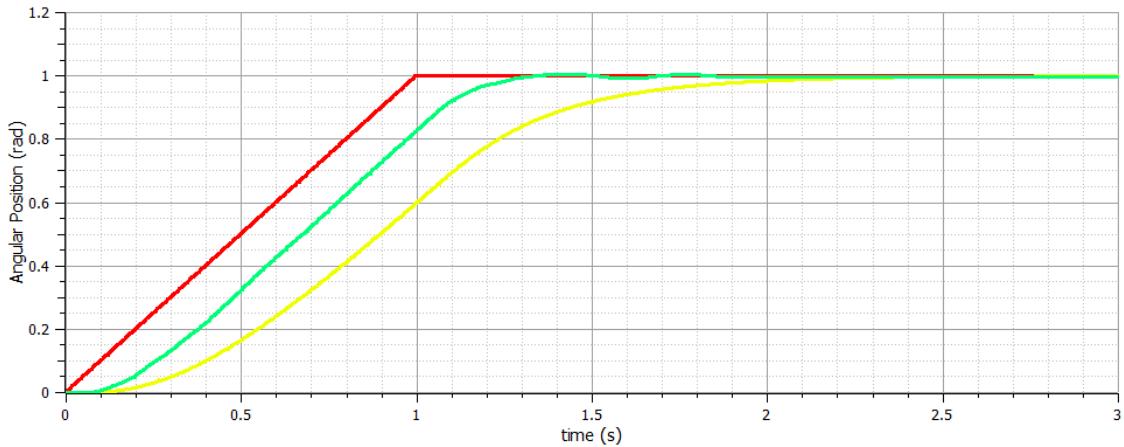


Figura 45. Respuesta a rampa en los tres ángulos. Psi

- Respuesta del control con realimentación taquimétrica a tren de pulsos en  $\varphi$ , sinusode en  $\theta$  y rampa en  $\psi$ . Referencia de  $\varphi$  en rojo, respuesta en amarillo. Referencia de  $\theta$  en fucsia, respuesta en azul oscuro. Referencia de  $\psi$  en verde, respuesta en naranja.

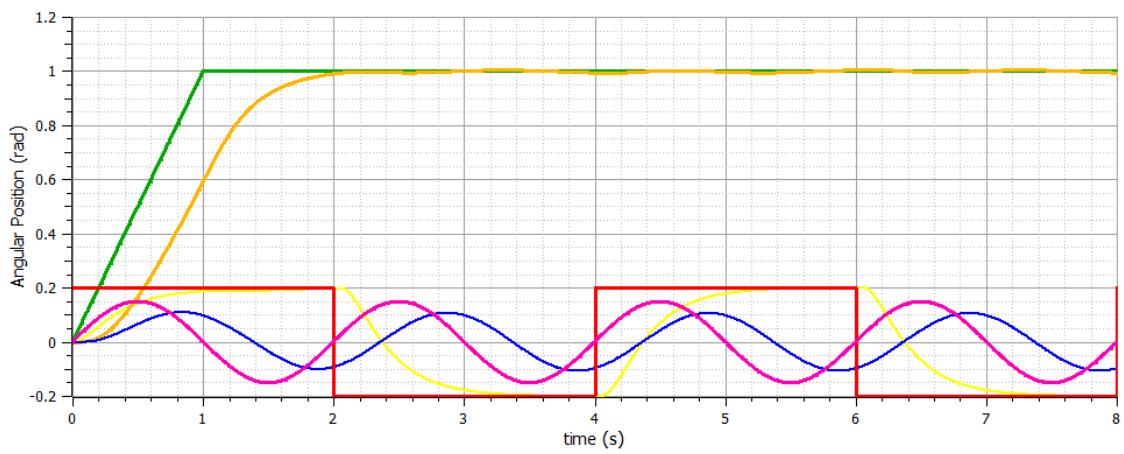


Figura 46. Referencias de tren de pulsos (phi), sinusode (theta) y rampa (psi) y sus respuestas con realimentación taquimétrica

Respuesta del control deadbeat a las mismas referencias anteriores. Referencia de  $\varphi$  en rojo, respuesta en amarillo. Referencia de  $\theta$  en fucsia, respuesta en azul oscuro. Referencia de  $\psi$  en verde, respuesta en negro.

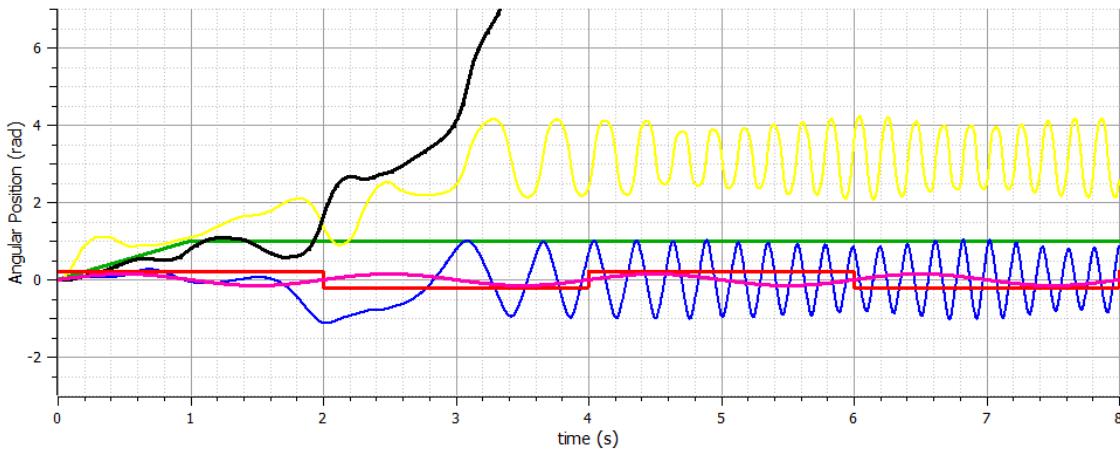


Figura 47. Referencias de tren de pulsos ( $\phi$ ), sinusoides ( $\theta$ ) y rampa ( $\psi$ ) y sus respuestas con deadbeat

Atendiendo a las gráficas de las figuras 38, 41 y 42, se podría decir que el control deadbeat funciona mucho mejor que el control por realimentación taquimétrica, ya que, aunque presenta un error de posición (casi nulo) y una sobreoscilación (pequeña y admisible), el tiempo de respuesta es mucho menor, debido a las grandes acciones que presenta (figuras 40 y 43). Por tanto, se diría que la respuesta presenta una dinámica más deseable en el caso del control deadbeat. Y esto es así cuando, la consigna solo cambia en una de las tres posiciones angulares.

Sin embargo, observando la gráfica de la figura 44, podemos ver que, al haber un cambio de referencia en los tres ángulos a la vez, la respuesta deadbeat para los ángulos  $\phi$  y  $\theta$  se empobrece, llegando a presentar un gran error de posición para  $\phi$ . Ocurre algo similar que con el PAF que intentamos implementar al principio. En las ecuaciones (49) y (50) que calculan las aceleraciones angulares en torno a los ejes X e Y del cuadricóptero, respectivamente, si las velocidades angulares en torno a los otros ejes no son nulas, las no linealidades pueden ser grandes.

Observando las figuras 46 y 47, en las que las referencias se parecen más a las consignas que podrían llegar de aguas arriba del control, vemos que con el control deadbeat las variables se vuelven directamente locas y no siguen para nada las referencias. En cambio, con la realimentación taquimétrica, aunque las respuestas tienen cierto retraso, el comportamiento es bastante satisfactorio.

Tomando como ejemplos las anteriores comparaciones de los distintos controles tanto para la regulación de las velocidades angulares de los motores como para las posiciones angulares (orientación) del cuadricóptero, podemos concluir que el control deadbeat es un muy buen control y hace que el sistema tenga muy buen comportamiento cuando el modelo de la planta a controlar es lo suficientemente fiel a

la realidad. Es el caso del control de la velocidad angular de los motores, cuyas ecuaciones que las rigen no tienen grandes no-linealidades ni ha sido necesario simplificar demasiado la planta para poder obtener un modelo manejable para diseñar el control. En cambio, cuando la planta se ha simplificado bastante o presenta fuertes no linealidades, el control deadbeat se muestra incapaz de llevar la respuesta cerca de la referencia y por el contrario la realimentación taquimétrica se muestra mejor.

### 3.3. Control de posición

Como en los casos anteriores, empezamos identificando qué hay que controlar y qué ecuaciones modelan su funcionamiento. Una vez calculado e implementado el control de velocidades de los rotores y el control de orientación del cuadricóptero, subimos un nivel más y controlaremos unas variables más cercanas al nivel usuario.

Estas variables son la posición en el espacio del centro de masas del cuadricóptero en la referencia global y sus derivadas las velocidades lineales y las aceleraciones lineales. El objetivo de este control será que estas variables sigan lo mejor posible las referencias que nos vendrán dadas por el último módulo, descrito a continuación del que nos ocupa: el generador de trayectorias.

Siguiendo el esquema de control global de la Figura 20, las variables de entrada a nuestro módulo de control de posición, es decir, nuestras referencias, son, como hemos dicho ya, las posiciones globales del cuadricóptero, y sus derivadas las velocidades y aceleraciones lineales. Las salidas de nuestro módulo son las siguientes: las posiciones angulares del cuadricóptero  $\varphi$ ,  $\theta$  y la fuerza de empuje  $T$ . Tanto  $\varphi$  como  $\theta$  serán entradas del siguiente módulo: el control de orientación. El ángulo  $\psi$ , entrada también del control de orientación, viene dado como entrada global de todo nuestro sistema de control. Para simplificar consideraremos esta entrada igual a cero.

La salida restante,  $T$ , permite mantener el cuadricóptero flotando y moverse en una determinada dirección dado el ángulo del cuadrirrotor adecuado. Tal como hemos dicho en la introducción del control de orientación, esta variable, entrada del módulo conversor de  $T$  y pares a velocidades angulares de los rotores, viene determinada por el control de posición.

Para diseñar este control nos hemos basado en el que se realiza en [4], determinado por las siguientes ecuaciones:

$$d_x = K_{x,P}(x_d - x) + K_{x,D}(\dot{x}_d - \dot{x}) + K_{x,DD}(\ddot{x}_d - \ddot{x}) \quad (66)$$

$$d_y = K_{y,P}(y_d - y) + K_{y,D}(\dot{y}_d - \dot{y}) + K_{y,DD}(\ddot{y}_d - \ddot{y}) \quad (67)$$

$$d_z = K_{z,P}(z_d - z) + K_{z,D}(\dot{z}_d - \dot{z}) + K_{z,DD}(\ddot{z}_d - \ddot{z}) \quad (68)$$

$$\varphi = \arcsen \left( \frac{d_x * S_\psi - d_y * C_\psi}{d_x^2 + d_y^2 + (d_z + g)^2} \right) \quad (69)$$

$$\theta = \arctg \left( \frac{d_x * C_\psi + d_y * S_\psi}{d_z + g} \right) \quad (70)$$

$$T = M_t * [d_x * (S_\theta * C_\psi * C_\varphi + S_\psi * S_\varphi) + \\ + d_y * (S_\theta * S_\psi * C_\varphi - C_\psi * S_\varphi) + (d_z + g) * C_\theta * C_\varphi] \quad (71)$$

Siendo  $S_\theta = \sin(\theta)$  y  $C_\theta = \cos(\theta)$ ,  $x_d$  la referencia de posición en el eje X y  $x$  la posición en el eje X medida.

Como disponemos tanto de las velocidades lineales como de las aceleraciones lineales medidas directamente del cuadricóptero, podemos implementar las ecuaciones (66), (67) y (68) como control de posición sin necesidad de discretizar, ya que solamente hay sumas de variables y multiplicaciones de variables por escalares. Las ecuaciones (69), (70) y (71) sirven de cálculos para generar las salidas de  $\varphi$ ,  $\theta$  y  $T$ , que serán usadas como consignas aguas abajo del control.

Hemos determinado que los valores de las constantes de este control más adecuados y que resultan en un mejor comportamiento del cuadricóptero son:

$$K_{x,P} = 1, K_{x,D} = 12, K_{x,DD} = 2$$

$$K_{y,P} = 40, K_{y,D} = 40, K_{y,DD} = 10$$

$$K_{z,P} = 20, K_{z,D} = 10, K_{z,DD} = 2$$

Los resultados de la implementación de este control de posición los expondremos después de explicar el último módulo: el generador de trayectorias.

### 3.4. Generador de trayectorias

El objetivo principal de todo el sistema de control del cuadricóptero es poder llevarlo de un punto del espacio a otro, con la mayor exactitud posible. Con el fin de llevar a cabo esta tarea de manera suave, sin cambios bruscos de velocidad y de aceleración, que puedan llevar a desestabilizar el robot o a picos de referencias en los

controles internos, hemos desarrollado un generador de trayectorias basado en el diseñado por [4].

La mecánica en la que se basa este sistema consiste en establecer una sinusode definida a tramos en la cuarta derivada de la posición, llamada chasquido o *jounce* en inglés. Esta sinusode depende de una serie de parámetros que permiten controlar la magnitud de la aceleración o cuánto dura dicha aceleración hasta alcanzar una “velocidad crucero” constante que durará todo el trayecto del cuadricóptero antes de decelerar y parar en el punto del espacio deseado. Esta función del “chasquido” a lo largo del tiempo será integrada sucesivamente para generar un perfil de tirón o sobreaceleración (tercera derivada de la posición), de la aceleración, de la velocidad y de la posición. Este proceso se llevará a cabo por separado para cada uno de los tres ejes espaciales, dado que los requerimientos de la posición y sus derivadas serán diferentes en un eje que en otro.

La función que define el chasquido es la siguiente:

$$f_c(t) = \begin{cases} a * \operatorname{sen}\left(\frac{1}{b} * \pi * t\right) & \text{para } 0 < t \leq b \\ -a * \operatorname{sen}\left(\frac{1}{2 * b} * \pi * t - \frac{\pi}{2}\right) & \text{para } b < t \leq 3 * b \\ a * \operatorname{sen}\left(\frac{1}{b} * \pi * t - 3 * \pi\right) & \text{para } 3 * b < t < 4 * b \\ 0 & \text{para } 4 * b \leq t \leq 4 * b + c \\ -a * \operatorname{sen}\left(\frac{1}{b} * \pi * (t - (4 * b + c))\right) & \text{para } 4 * b + c < t \leq 5 * b + c \\ a * \operatorname{sen}\left(\frac{1}{2 * b} * \pi * (t - (4 * b + c)) - \frac{\pi}{2}\right) & \text{para } 5 * b + c < t \leq 7 * b + c \\ -a * \operatorname{sen}\left(\frac{1}{b} * \pi * (t - (4 * b + c)) - 3 * \pi\right) & \text{para } 7 * b + c < t \leq 8 * b + c \\ 0 & \text{para } t > 8 * b + c \end{cases} \quad (72)$$

Esta función básicamente consiste en los tramos de sinusode de la figura 48 para acelerar, un tramo de tiempo  $c$  con valor 0 y la función negativa de dicha figura para decelerar. Como se ve en dicha figura, el valor de  $a$  indica la amplitud de la sinusode, es decir, cuán rápido acelera el cuadricóptero;  $b$  es la mitad de un periodo de sinusode y determinará durante cuánto tiempo está acelerando el robot. El parámetro  $c$  es el tiempo en que la velocidad es constante tras terminar de acelerar y antes de empezar a decelerar.

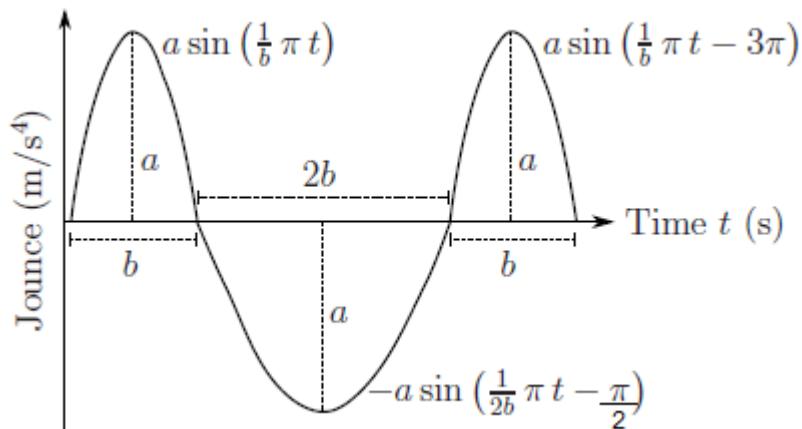


Figura 48. Función del chasquido. Tomada de [4]

Vamos a proceder a dar valor a estos parámetros. En nuestro caso hemos fijado el valor de  $b$  a 0.25s. Para calcular el valor de  $c$  primero debemos establecer la velocidad máxima, que será la velocidad de crucero en un eje del espacio. Hemos decidido  $V_{\max} = 0.3 \text{ m/s}$ . Queremos que las referencias de posición generadas (que se asemejarán a rampas) lleguen al valor deseado a la vez en los tres ejes del espacio, de lo contrario el cuadricóptero tendría un movimiento extraño primero avanzando en los tres ejes a la vez para al final avanzar en solo un eje del espacio y llegar al punto de destino. Por tanto, será el eje cuya consigna sea mayor la que determine el valor de  $c$  y el cual tendrá asignada la  $V_{\max}$ . Los demás ejes, con menor distancia que recorrer, tendrán una velocidad crucero menor para poder terminar los tres a la vez. El valor de  $c$  viene determinado por la siguiente expresión, tomando como ejemplo que la mayor consigna de posición es en el eje X:

$$c = \frac{|x_d| - V_{\max}}{V_{\max}} \quad (73)$$

Los valores de las velocidades máximas serán entonces:

$$V_{\max,x} = \frac{x_d}{c + 1} \quad (74)$$

$$V_{\max,y} = \frac{y_d}{c + 1} \quad (75)$$

$$V_{\max,z} = \frac{z_d}{c + 1} \quad (76)$$

El último parámetro por determinar es la  $a$ :

$$a_x = \frac{V_{\max,x}}{0.02097} \quad (77)$$

$$a_y = \frac{V_{máx,y}}{0.02097} \quad (78)$$

$$a_z = \frac{V_{máx,z}}{0.02097} \quad (79)$$

Ahora ya disponemos de lo necesario para implementar la función del chasquido e integrarla para generar las trayectorias. Estos son los resultados para una referencia de 1.5m en el eje Y.

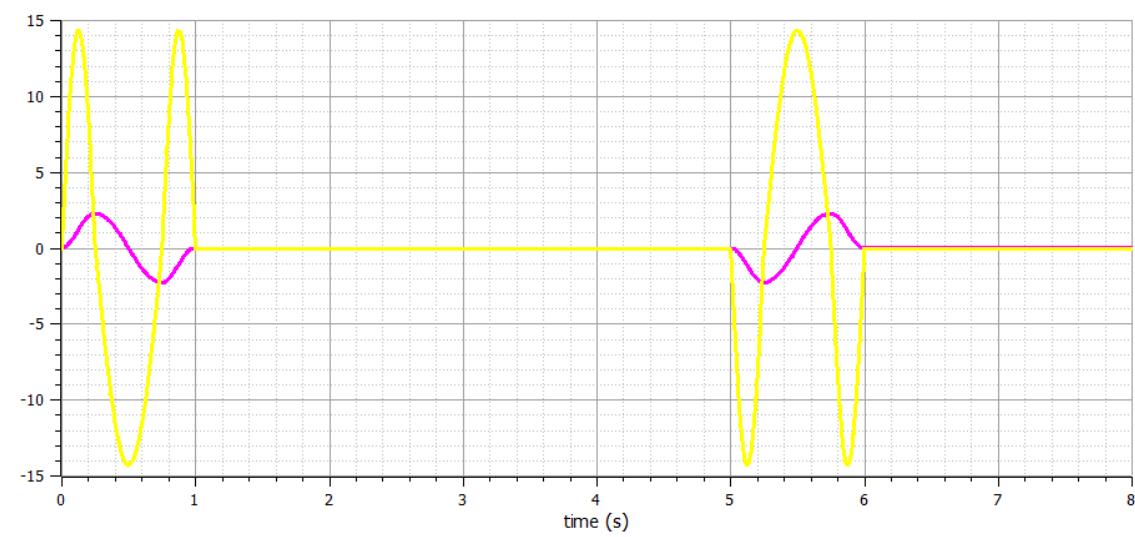


Figura 49. Chasquido (jounce) y tirón (jerk)

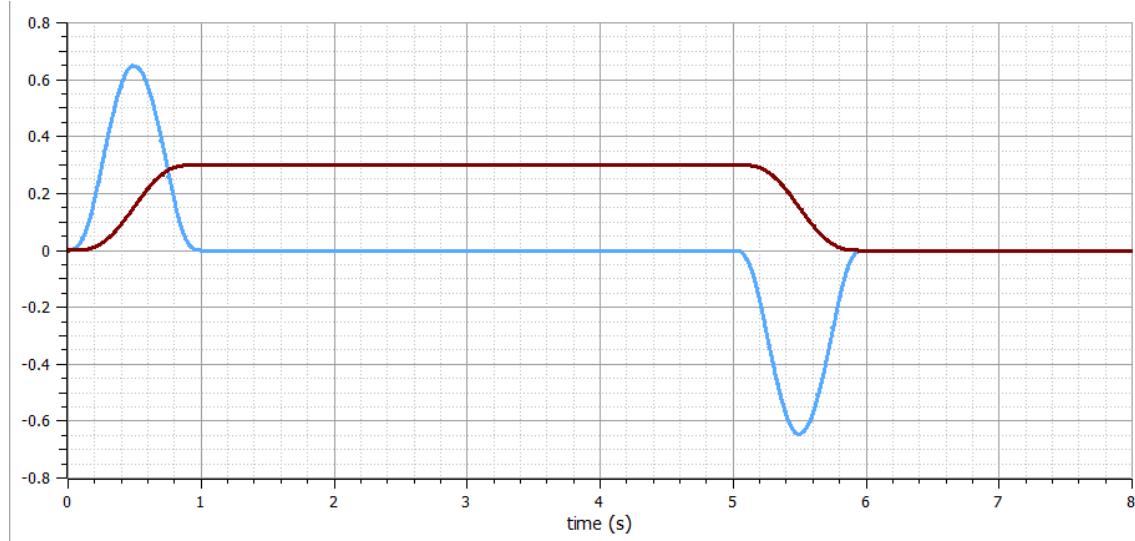


Figura 50. Aceleración y velocidad

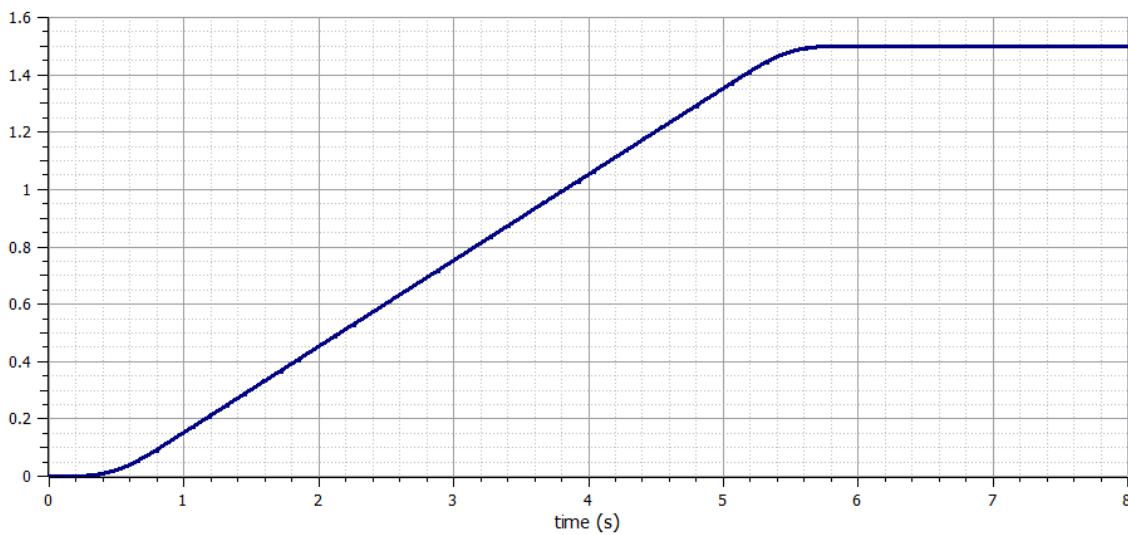


Figura 51. Posición

Como podemos observar, cada gráfica se obtiene integrando la anterior, vemos una velocidad máxima constante de 0.3 m/s y cómo la gráfica de la posición llega hasta 1.5m. Estos serán los perfiles de posición, velocidad y aceleración para cada eje que se introducen como consignas en el control de posición. Así pues, como mencionamos antes, vamos a presentar ahora los resultados del control de posición, ahora que disponemos de unas referencias de trayectoria suaves y satisfactorias. Con una referencia de X = 3m, Y = 2m y Z = 1m:

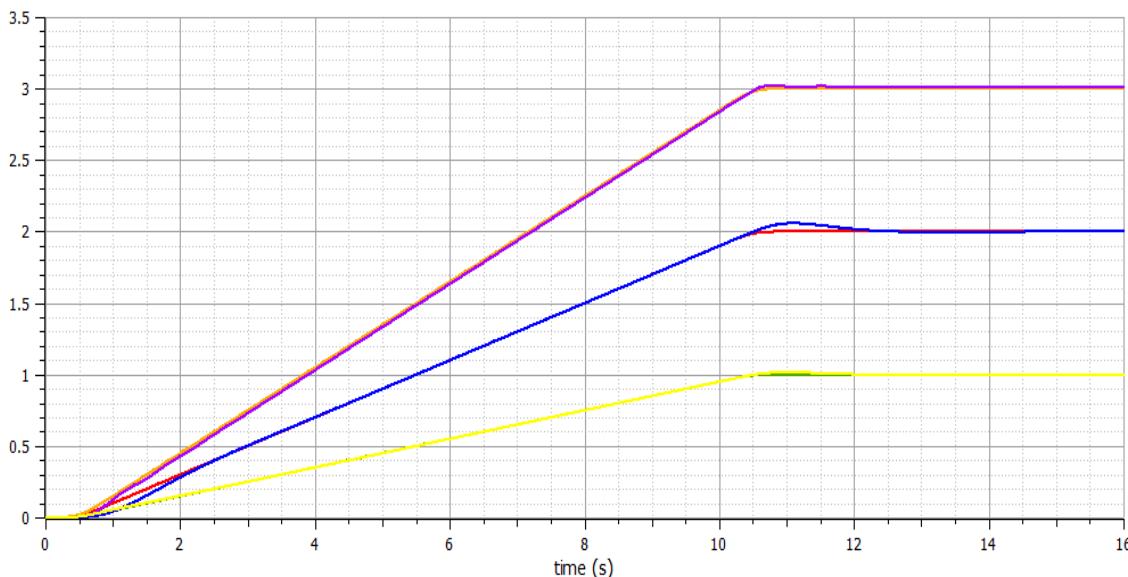


Figura 52. Referencias de posición en X (naranja), en Y (rojo) y en Z (verde) y posición real en X (violeta), en Y (azul) y en Z (amarillo)

Como podemos apreciar en la Figura 52, el generador de trayectorias se ha encargado de que las tres rampas de posición lleguen al destino a la vez. Asimismo, el

control de posición consigue que el cuadricóptero siga satisfactoriamente la referencia de posición dada por el generador de trayectorias. El objetivo global de nuestro control ha sido cumplido y hemos sido capaces de llevar a nuestro vehículo de un punto del espacio a otro con la suficiente precisión.

## 4. Conclusiones

Hemos realizado en primer lugar un modelo de un cuadricóptero con disposición en cruz, tanto de su movimiento (cinemática) como de las fuerzas que actúan en él (dinámica) y del funcionamiento de los motores de corriente continua para las hélices. Hemos verificado que este modelo se comportaba como queríamos y como esperábamos mediante una serie de experimentos.

Seguidamente hemos procedido a diseñar el control completo del cuadricóptero, comenzando por el control de las velocidades angulares de las hélices, siguiendo con el control de orientación y finalmente realizando el control de posición. Además, se ha elaborado un generador de trayectorias para disponer de referencias suaves. Tanto con el primer control como con el segundo, se han utilizado dos métodos distintos, se han realizado experimentos, se han comparado sus resultados y se ha llegado a unas conclusiones correspondientes.

Estas conclusiones se pueden sintetizar en cómo el control deadbeat exhibe un desempeño realmente bueno y rápido cuando el modelo de nuestra planta es suficientemente fiel a la realidad y no existen grandes no-linealidades, pero cómo su funcionamiento es pobre o inadecuado cuando estas condiciones no se dan. Asimismo, un control por compensación de polo lento y posterior discretización por emulación, puede ser relativamente lento en comparación con otros controles como el deadbeat.

Por otro lado, la realimentación taquimétrica se ha mostrado totalmente adecuada y resolutiva cuando otros controles, como el deadbeat o un proporcional avance de fase fracasan, como en el control de orientación.

En cuanto a las futuras líneas a trabajar o pasos a seguir, mostramos unos cuantos: se podría realizar el diseño de un control en espacio de estados que tenga como entradas las posiciones globales y psi y como acciones la fuerza T y los pares, de manera que sustituya a los controles aquí realizados de orientación y posición, dejando intactos el control de velocidad de los motores y el módulo de conversión de fuerzas y pares a velocidades angulares. También se podría diseñar un generador de trayectorias más sofisticado que permita pasar por un determinado punto con una velocidad determinada para encaminarse hacia un nuevo punto del espacio y así sucesivamente. Además, se puede incluir en el modelo la resistencia aerodinámica que sufre el cuadricóptero cuando se mueve. Igualmente, es posible incluir un modelado de los distintos sensores que puede haber para medir las variables de interés, como un IMU para las aceleraciones lineales, la orientación y las velocidades angulares, un GPS para

## Modelado, simulación y control de cuadricópteros

Enrique Cañada Panea

las posiciones en X e Y, y un sensor de presión atmosférica para posición en Z. Estos sensores pueden incluir cierto ruido, como por ejemplo un offset y una pequeña sinusoide o un ruido aleatorio. Por tanto, para estimar mejor las medidas se podría probar utilizar un filtro de Kalman.

Respecto a lo aprendido con este trabajo, se podría destacar el control con realimentación taquimétrica; la implementación de control deadbeat, que solo había visto de manera teórica, o el recurrir a terceras y cuartas derivadas de la posición para crear perfiles de trayectoria suaves. También concebir y elaborar un proyecto tan grande, extenso y complejo.

## 5. Referencias

- [1] E. Kofman, "Introducción a Modelica."
- [2] P. Fritzson, *Introducción al Modelado y Simulación de Sistemas Técnicos y Físicos con Modelica*. 2003.
- [3] M. Kuric, N. Osmic, and A. Tahirovic, "Multirotor Aerial Vehicle modeling in Modelica," in *Proceedings of the 12th International Modelica Conference, Prague, Czech Republic, May 15-17, 2017*, 2017, vol. 132, pp. 373–380.
- [4] T. Luukkonen, "Modelling and control of quadcopter," Espoo, 2011.
- [5] D. J. Quiñones Guerra, "Control de actitud de un cuadricóptero," 2018.
- [6] R. Mahony, V. Kumar, and P. Corke, "Multirotor aerial vehicles: Modeling, estimation, and control of quadrotor," *IEEE Robot. Autom. Mag.*, vol. 19, no. 3, pp. 20–32, 2012.
- [7] M. D. Schmidt, "SIMULATION AND CONTROL OF A QUADROTOR UNMANNED AERIAL VEHICLE," Kentucky, 2011.
- [8] T. Tengis and A. Batmunkh, *State feedback control simulation of quadcopter model*. Novosibirsk, Russia: 11th International Forum on Strategic Technology (IFOST), 2016.
- [9] S. Bouabdallah and R. Siegwart, "Full Control of a Quadrotor," 2009.
- [10] L. E. Romero Paredes, "Diseño y construcción de un módulo de control PID para la estabilización de un cuadricóptero," Quito, 2014.
- [11] L. E. Romero, D. F. Pozo, and J. A. Rosales, "Quadcopter stabilization by using PID controllers," 2014.
- [12] M. Dursun and S. Engin, "Deadbeat control of a DC servo motor at low speed," in *Proceedings - 2018 4th International Conference on Control, Automation and Robotics, ICCAR 2018*, 2018, pp. 282–286.

## 6. Tabla de figuras

Figura 1. Rotores en cruz y en equis. Vista de planta. Tomado de [8].	5
Figura 2. Giro de los rotores y definición de los ejes respecto de los números de los rotores. Tomado de [3].	6
Figura 3. Ejes de coordenadas globales y locales. Tomado de [3]	7
Figura 4. Roll, pitch y yaw. Tomada de [8].	8
Figura 5. Giro de los ejes. Tomada de [7].	8
Figura 6. Velocidades lineales y angulares en las coordenadas locales. Tomado de [3].	9
Figura 7. Fuerzas y pares principales. Tomado de [3].	10
Figura 8. Fuerzas de empuje producidas por los rotores. Tomado de [3].	11
Figura 9. Masas y dimensiones del cuadricóptero. Adaptada de [3].	13
Figura 10. Velocidades angulares de los rotores con referencia de 600 rad/s en todos ellos.	18
Figura 11. Posición en Z (rojo) y aceleración en Z (azul) con referencia de 600 rad/s en todos los rotores.	18
Figura 12. Velocidades angulares de los rotores con referencias de $\Omega_1 = 700$ , $\Omega_2 = \Omega_4 = 600$ , $\Omega_3 = 479.78$ rad/s	19
Figura 13. Ángulo phi con referencias de $\Omega_1 = 700$ , $\Omega_2 = \Omega_4 = 600$ , $\Omega_3 = 479.78$ rad/s	19
Figura 14. Posición en Y (rojo) y par en el eje X (azul) con referencias de $\Omega_1 = 700$ , $\Omega_2 = \Omega_4 = 600$ , $\Omega_3 = 479.78$ rad/s	20
Figura 15. Velocidades angulares de los rotores con referencias de $\Omega_2 = 800$ , $\Omega_1 = \Omega_3 = 600$ , $\Omega_4 = 282.84$ rad/s	20
Figura 16. Ángulo theta con referencias de $\Omega_2 = 800$ , $\Omega_1 = \Omega_3 = 600$ , $\Omega_4 = 282.84$ rad/s	21
Figura 17. Posición en X (rojo) y par en el eje Y (azul) con referencias de $\Omega_2 = 800$ , $\Omega_1 = \Omega_3 = 600$ , $\Omega_4 = 282.84$ rad/s	21
Figura 18. Velocidades angulares de los rotores con referencias de $\Omega_2 = \Omega_4 = 600$ , $\Omega_1 = \Omega_3 = 700$ rad/s	22
Figura 19. Posición en Z (rojo), par en el eje Z (azul) y ángulo psi (verde) con referencias de $\Omega_2 = \Omega_4 = 600$ , $\Omega_1 = \Omega_3 = 700$ rad/s	22
Figura 20. Esquema de control completo	23
Figura 21. Respuesta al escalón de 10V de un motor.	25
Figura 22. Lugar de las raíces del sistema $R(s)*G(s)$ .	27
Figura 23. Respuesta al escalón del sistema en bucle cerrado, continuo y discreto, en MATLAB.	28
Figura 24. Algoritmo de control de los motores por emulación	29
Figura 25. Esquema de control deadbeat. Adaptada de [12]	29
Figura 26. Algoritmo de control deadbeat de los motores	30
Figura 27. Respuesta al escalón (rosa). Velocidades angulares. Emulación (rojo) y deadbeat (verde)	31
Figura 28. Respuesta al escalón. Acciones. Emulación (verde) y deadbeat (azul)	31
Figura 29. Respuesta a la rampa (amarillo). Velocidades angulares. Emulación (rojo) y deadbeat (verde)	32

Figura 30. Respuesta a la rampa. Acciones. Emulación (morado) y deadbeat (azul oscuro) .....	32
Figura 31. Respuesta a un tren de pulsos (morado). Velocidades angulares. Emulación (naranja) y deadbeat (turquesa).....	32
Figura 32. Respuesta a un tren de pulsos. Acciones. Emulación (rosa) y deadbeat (morado) .....	33
Figura 33. Respuesta a una sinusoide (fucsia). Velocidades angulares. Emulación (amarillo) y deadbeat (celeste).....	33
Figura 34. Respuesta a una sinusoide. Acciones. Emulación (verde) y deadbeat (azul oscuro) .....	33
Figura 35. Ecuación de movimiento angular tomada de [4]. .....	36
Figura 36. Esquema de control por realimentación taquimétrica .....	38
Figura 37. Algoritmo de control por realimentación taquimétrica .....	39
Figura 38. Referencia de rampa en phi (naranja). Respuesta deadbeat (verde) y realimentación taquimétrica (amarillo) .....	42
Figura 39. Par ejercido en el eje X con realimentación taquimétrica .....	42
Figura 40. Pares en el eje X con deadbeat (rojo) y realimentación taquimétrica (azul) .....	42
Figura 41. Referencia de rampa en theta (fucsia). Respuesta deadbeat (amarillo) y realimentación taquimétrica (verde) .....	43
Figura 42. Referencia de rampa en psi (rojo). Respuesta deadbeat (verde) y realimentación taquimétrica (amarillo) .....	43
Figura 43. Pares en el eje Z con deadbeat (azul) y realimentación taquimétrica (morado).....	44
Figura 44. Respuesta a rampas en los tres ángulos. Phi y theta .....	44
Figura 45. Respuesta a rampa en los tres ángulos. Psi.....	45
Figura 46. Referencias de tren de pulsos (phi), sinusoide (theta) y rampa (psi) y sus respuestas con realimentación taquimétrica.....	45
Figura 47. Referencias de tren de pulsos (phi), sinusoide (theta) y rampa (psi) y sus respuestas con deadbeat .....	46
Figura 48. Función del chasquido. Tomada de [4] .....	50
Figura 49. Chasquido (jounce) y tirón (jerk) .....	51
Figura 50. Aceleración y velocidad .....	51
Figura 51. Posición .....	52
Figura 52. Referencias de posición en X (naranja), en Y (rojo) y en Z (verde) y posición real en X (violeta), en Y (azul) y en Z (amarillo).....	52