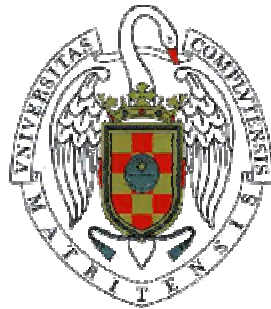


Control para la estabilización en suspensión de un cuatrimotor



Marta Salas García

Directores: José A. López Orozco
Jesús M. de la Cruz

Departamento de Arquitectura de Computadores y Automática

Facultad de Ciencias Físicas

Septiembre 2010

ÍNDICE

1. INTRODUCCIÓN	1
1.1. Objetivos	2
1.2. Estado del arte	2
1.2.1. Universidad Estatal de Pensilvania.....	3
1.2.2. Universidad de la Columbia Británica en Vancouver, BC, Canadá.....	3
1.2.3. Universidad Nacional de Australia.....	4
1.2.4. Universidad de Cornell.....	5
1.2.5. Instituto Federal Tecnológico Suizo	6
1.2.6. Universidad de Stanford	7
2. MODELADO Y ANÁLISIS DE LOS MOTORES.....	9
2.1. Descripción del sistema.....	10
2.1.1. Dinámica del sistema	10
2.1.2. Sensor de altura	11
2.2. Cálculo de la función de transferencia	12
2.2.1. Toma de datos.....	12
2.2.2. Función de transferencia	13
2.3. Verificación de la función de transferencia.....	14
2.4. Comparación del sistema físico con el modelo extraído	14
3. CONTROL DEL CUATRIMOTOR	17
3.1. Simplificación del modelo	17
3.2. Control LQR	18
3.2.1. Linealización por realimentación de estados.....	19
3.2.2. Controlador LQR	22
3.3. Control por modos deslizantes.....	25
3.4. Comparación de los controles	28
3.5. Control en posición del cuatrimotor.....	29
4. PRUEBAS REALES.....	33
4.1. Torre 3D Hover de Quanser.....	33
4.1.1. Realimentación de estados y controlador LQR.....	34
4.1.2. Control por modos deslizantes	38
5. CONCLUSIONES Y FUTUROS TRABAJOS.....	41
5.1. Conclusiones	41
5.2. Trabajo futuro	41
6. BIBLIOGRAFÍA	43
APÉNDICE A. CINEMÁTICA Y DINÁMICA DE UN CUATRIMOTOR	I
A. Cinemática.....	i
B. Dinámica.....	ii

1. INTRODUCCIÓN

Debido a los enormes avances en las técnicas de fabricación e innovación en conocimientos de los materiales, actualmente se fabrican sensores mucho más pequeños y precisos. La tecnología de los sistemas microelectrónico-mecánicos (MEMs) permite actualmente la producción de componentes de máquinas tales como engranajes con tamaños de micrómetros. Usando esta tecnología MEMs se fabrican acelerómetros, giróscopos y magnetómetros más pequeños, con los cuales se elaboran sistemas de navegación inercial (IMU, por sus siglas en inglés, *inertial measurement unit*). También ha sido posible el desarrollo de los microcontroladores. Todos estos avances han permitido la implementación de plataformas experimentales para el desarrollo de sistemas autónomos.

En este trabajo de master centrará la atención en sistema aéreos no tripulados (UAVs, por sus siglas en inglés, *unmanned aerial vehicle*) concretamente en helicópteros con cuatro rotores, cuatrimotores, estos tienen algunas ventajas sobre los helicópteros convencionales, ya que no es necesario un rotor de cola para anular la resistencia del rotor principal. El par en la dirección de orientación se cancela al girar dos de las cuatro hélices en el sentido de las agujas del reloj y las otras dos en el sentido contrario. El control de los ángulos de orientación: cabeceo, alabeo y guiñada, se ajusta variando la velocidad de la rotación de las hélices.

Dentro de sus aplicaciones civiles tenemos la vigilancia del tráfico de carreteras, operaciones de rescate y salvamento, recolección de información para la predicción meteorológica, vigilancia de bosques y detección de fuegos. También se puede destacar que pueden ser usados como plataforma tecnológica para la validación de conceptos aerodinámicos o leyes de control.

1.1. Objetivos

En el presente trabajo se desea controlar una plataforma experimental de este tipo en la que se pondrá en funcionamiento algoritmos de control, búsqueda y cooperación. El primer paso para el control de vuelo de un cuatrimotor será su estabilización en el aire.

Para la construcción de un cuatrimotor primeramente se llevará a cabo el análisis y estudio de los motores que se incorporarán en el vehículo, así como su estudio teórico y análisis de sus posibilidades. Ello permitirá ajustar el diseño y construcción de la plataforma a las necesidades del proyecto e identificar un modelo físico teórico que permita el diseño del controlador más adecuado.

Una vez construido el modelo teórico del cuatrimotor, desarrollado y probados en él los algoritmos de control para su estabilización; se procederá al control y estabilización del cuatrimotor construido.

1.2. Estado del arte

En 1907, los hermanos Breguet construyeron el primer cuatrimotor llamado Gyroplane No. 1 (Figura 1). El vuelo fue un buen trabajo para mostrar los principios de un cuatrimotor. En 1922, Georges de Bothezat construyó un cuatrimotor con un rotor localizado en cada extremo de unas vigas que formaban un armazón en forma de cruz. [Gordon, 2002]

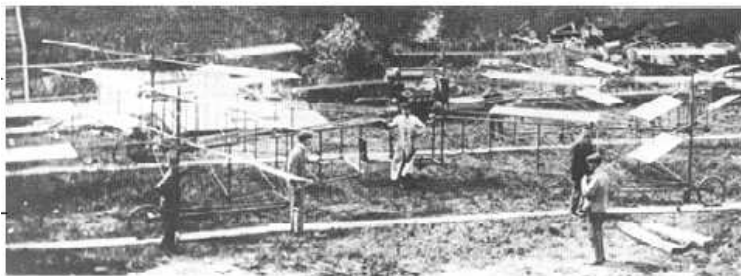


Figura 1. Cuatrimotor de Bréguet-Richet Gyroplane No. 1

Actualmente en el mercado podemos encontrar algunos cuatrimotores como el DraganFlyer (Figura 2). Recientemente en la literatura podemos encontrar diferentes estudios sobre cuatrimotores. Estos trabajos utilizan diferentes controladores, equipos y materiales, a continuación se ha hecho una recopilación de los estudios en diferentes universidades y centros de investigación.



Figura 2. DraganFlyer

1.2.1. Universidad Estatal de Pensilvania

En la universidad estatal de Pensilvania han hecho dos estudios diferentes sobre cuatrimotores. El primero de ellos es una tesis [Hanford, 2005] (Figura 3) en el que se emplea un cuatrimotor como banco de pruebas. La unidad de medida inercial consiste en tres giróscopos analógicos (ADXRS150EB) y un acelerómetro (ADXL210EB). Este cuatrimotor está controlado por un regulador tipo PI.

El segundo estudio realizado en Pensilvania (Figura 4) utiliza un DraganFlyer como sistema experimental. Lleva a bordo un giroscopio de tres ejes y una cámara. Una cámara situada en la tierra captura el movimiento de 5 marcas coloreadas de 2.5 cm presentes debajo del DraganFlyer, para obtener los ángulos de cabeceo, alabeo y guiñada y la posición del cuatrimotor utilizando el algoritmo tracking y una rutina de conversión. En otras palabras, el método de las dos cámaras se ha usado para estimar los seis grados de libertad de la posición del helicóptero. Debido a la limitación de peso no se pudieron añadir al sistema un GPS u otros acelerómetros. El controlador obtiene la posición y velocidad relativas solamente de las cámaras.

Se han probado dos métodos de control: uno usando una serie de mode-based, controlador por realimentación linealizada y el otro usa una ley de control back-stepping. El helicóptero está restringido con una atadura en la vertical, permitiendo el movimiento de guiñada y limitando las translaciones en x e y. Se han realizado simulaciones en MATLAB-Simulink que muestran la habilidad del controlador, incluso cuando hay errores en los estados estimados.

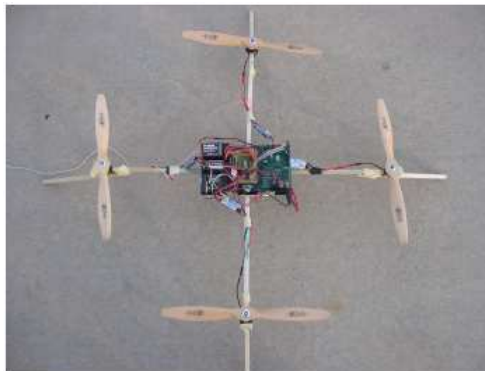


Figura 3. Cuatrimotor diseñado en la universidad de Pensilvania

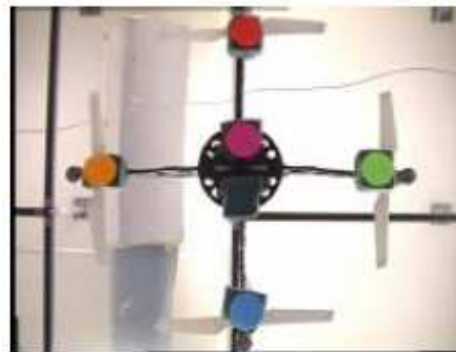


Figura 4. Cuatrimotor con cámara para prueba del algoritmo de tracking

1.2.2. Universidad de la Columbia Británica en Vancouver, BC, Canadá.

El sistema fue desarrollado en el Departamento de Ingeniería de computadores y electricidad de Universidad Británica de Columbia en Vancouver, BC, Canadá [Chen, 2003]. Este trabajo se centra en el modelado no lineal de un cuatrimotor. Para hacer prueba con el controlador se ha usado un

sistema experimental que incluye: un *molino volador*¹, un procesador digital de señales o DSP (por sus siglas en inglés, *Digital Signal Processor*), un microprocesador programado y un transmisor inalámbrico. Basado en el modelo no lineal, se ha diseñado un lazo de control H_{∞} para estabilización y el control de la velocidad, de la potencia y de la guiñada. El controlador desarrollado utilizando la técnica de control predictivo basado en modelos MBPC (por sus siglas en inglés, *Model Based Predictive Control*) y se implementa para el control de la trayectoria lateral y longitudinal.

Se ha programado un microprocesador, PIC16F877, para interpretar los datos de control a señal PWM (por sus siglas en inglés, *Pulse Width Modulated*, Modulación por Anchura de Pulso) con el propósito de una significativa reducción de carga de CPU; de otro modo habría que haberlo asociado con un DS1102. Esta señal es además usada para controlar los cuatro rotores del Draganflyer II a través de un transmisor de radio Futaba de 4 canales trabajando en modo entrenador.

Con el fin de realizar experimentos de control de vuelo se construyó una plataforma experimental que incluía un *molino volador* diseñado a mediada, un ordenador personal, una placa dSPACE DSP, un microprocesador modulador de pulsos, un transmisor de radio y el Draganflyer III (Figura 5).

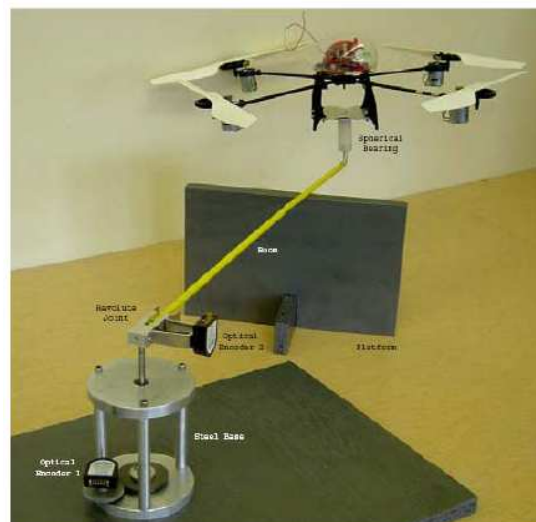


Figura 5. *Molino volador* desarrollado por la Universidad Británica de Columbia en Vancouver

1.2.3. Universidad Nacional de Australia

El X4-Flyer desarrollado en la Universidad Nacional de Australia consiste un computador HC-12 a bordo desarrollado por QCAT que se usa como un sistema de señal condicionado [Pounds *et al*, 2002a] (Figura 6). Esta tarjeta usa dos procesadores HC-12, señales de salida PWM que controlan directamente la velocidad, señales de entrada PWM que se reciben en un receptor RC R700 JR Slimline permitiendo pilotar el sistema desde una emisora JP 3810 y tiene dos canales serie RS232

¹NT. Flying mill traducido literalmente por molino volador, se refiere a la plataforma experimental de prueba creada por la Universidad de la Columbia Británica y se puede ver en la Figura 5

separados, el primero se usa como interfaz con la IMU y el segundo se usa como un link de datos asíncrono con el ordenador base en tierra. El controlador de velocidad empleado es un MSC30 B con peso de 26g y alimentado con 30A y 12V.

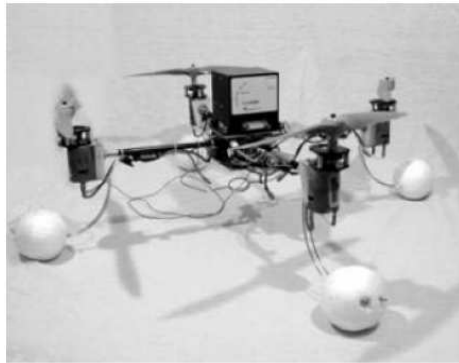


Figura 6. X4-Flyer desarrollado en la Universidad Nacional de Australia

1.2.4. Universidad de Cornell

El proyecto de la Universidad de Cornell consiste en crear un vehículo autónomo fiable, capaz de permanecer estable en el aire [Brian, 2004]. En el sistema de empuje se han usado motores MaxCim. El vehículo final pesa aproximadamente 6.22Kg.

El vehículo fue diseñado como un sistema independiente con alimentación abordo, unidad de control, comunicaciones inalámbricas y procesador. El sistema además está instrumentado con una unidad de medida inercial de seis grados de libertad y un sistema de posicionamiento inercial interior. La IMU mide las aceleraciones y las tasas de cambio de las rotaciones con una tasa de actualización alta (100Hz), mientras que el GPS proporciona datos de la posición en tres dimensiones a baja tasa (5Hz).

Con el propósito de realizar un control de navegación fiable y robusto, han desarrollado un estimador para fusionar los dos flujos de datos y obtener una estimación del estado del vehículo. El problema de estimación del estado completo es el alto acoplamiento no lineal.

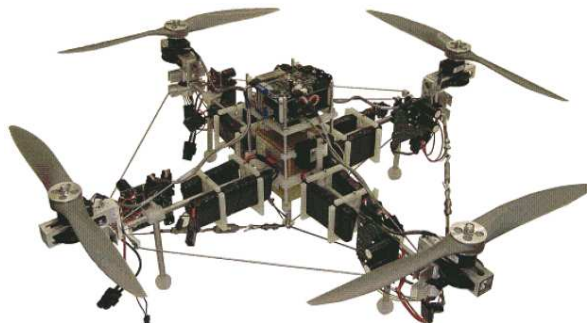


Figura 7. Cuatrimotor diseñado por la Universidad de Cornell

1.2.5. Instituto Federal Tecnológico Suizo

El instituto Federal Tecnológico Suizo ha llevado a cabo el estudio en el diseño mecánico, modelado dinámico, dotación de sensores y control de un VTOL autónomo en interior del robot OS43. Restringido a tres grados de libertad. [Bouabdallah *et al*, 2004]

Desde un PC y a través de un puerto estándar RS232, se mandan órdenes al banco de pruebas. Un módulo RS232 a I2C se encarga de traducir las señales del puerto serie al módulo de los motores I2C. Este módulo integra un regulador PID en un microcontrolador PIC16F876 y son capaces de abrir o cerrar el lazo de operación en el control de posición, velocidad y torque. El MT9-B8 IMU9 estima, con un filtro de Kalman, los datos de orientación en 3D y proporciona los datos de calibración de aceleración y velocidad angular. Esto pesa unos 33g y se comunica a 151Kbps. Con la ayuda de decodificadores micro-ópticos en cada eje se captura el movimiento de la articulación para extraer información de orientación absoluta.

La cruz está hecha con varillas de carbono, la masa de las cuales es de unos 240g, por lo tanto el vehículo es muy ligero. El banco de pruebas OS4 tiene cuatro grupos de propulsión, cada uno compuesto por un motor de 29g incluyendo encoders magnéticos, una caja de engranajes de 6g y una hélice de 6g.

Antes de la aplicación sobre el sistema real, se han realizado varias simulaciones en MATLAB. La tarea del controlador es estabilizar en altura mientras se compensa el error inicial en los ángulos de alabeo, cabeceo y guiñada. El sistema real sufre de, no deseados, pero inevitables retrasos y saturaciones del actuador. Los retardos se deben principalmente a las comunicaciones RS232 y a la constante de tiempo de actuación. Para simular estas carencias se han empleado dos bloques de retardo discreto de Simulink situados en el lazo de la realimentación y en los actuadores. El nivel de saturación depende de los actuadores elegidos. El montaje experimental se muestra en la Figura 8.

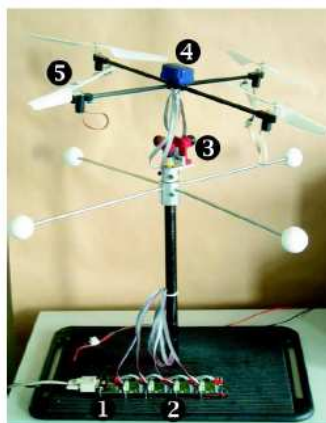


Figura 8. Cuatrimotor diseñado por el instituto Federal Tecnológico Suizo

1.2.6. Universidad de Stanford

STARMAC es el nombre del proyecto de la Universidad de Stanford [Hoffmann *et al*, 2004]. STARMAC consta de dos helicópteros X4-flyer que de forma autónoma pueden seguir una trayectoria con un punto de referencia dado, Figura 9. El algoritmo de planificación de trayectorias genera dicha trayectoria para sistemas multiagente. El proyecto STARMAC tiene como objetivo crear un sistema autónomo totalmente fiable capaz de rastrear puntos de referencia, por lo que es un banco de pruebas útil para los algoritmos de alto nivel dirigido a la coordinación de varios vehículos.

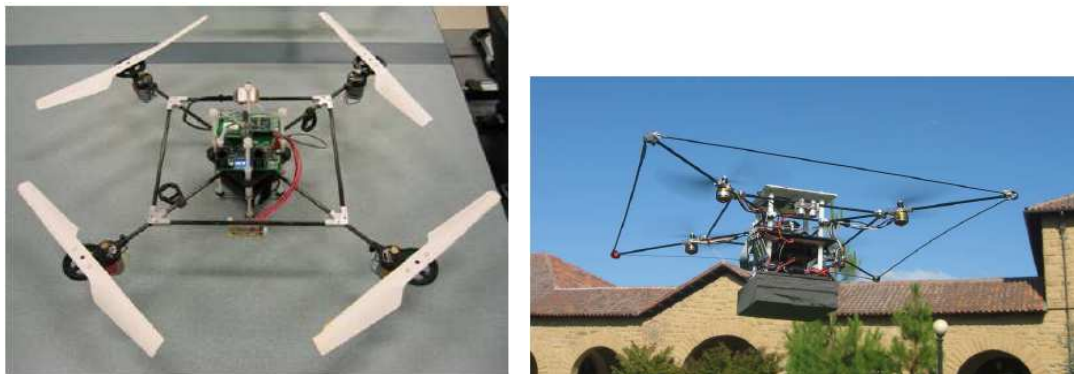


Figura 9. Cuatrimotor diseñado por la universidad de Stanford

El sistema base es el helicóptero de cuatro rotores DraganFlyer III, que puede levantar aproximadamente 113,40 gramos de carga útil y volar durante unos diez minutos a máxima potencia debido a una reciente actualización de las baterías de litio-polímero, las cuales han incrementado la carga útil y la duración del vuelo, y por lo tanto, ha mejorado considerablemente la capacidad del sistema.

Las unidades electrónicas a bordo son unas PCB diseñadas en Stanford, incorporan todos los sensores y las funciones de comunicación para realizar vuelos autónomos. Dos microcontroladores PIC18F6520 de Microchip coordinan las actividades de comunicación, detección, estimación y control a bordo de los cuatrimotores.

Los datos de vuelo son transmitidos a la estación de tierra y ésta envía las órdenes vía Bluetooth, dispositivo de clase II que tiene un alcance de unos 90 metros, aunque en la práctica, los datos comienzan a perderse a unos 45 metros. Este es un dispositivo distinto del que se emplea para usos domésticos, por lo que tiene mayor alcance. El dispositivo opera a 2.4 GHz e incorpora corrección de errores y retransmisión automática. Opera con un ancho de banda máximo de 115.2Kbps.

Los sensores que incluye a bordo son: una unidad GPS diferencial (Trimble Lassen LP), un altímetro de ultrasonidos (Devantech SRF08) al cual nos referiremos como *Sodar* y una IMU de tres ejes (MicroStrain 3DM-G) para detectar el estado del sistema. Las actualizaciones del GPS se producen a 1Hz, y tiene un rango de error diferencial estándar de 1-2m en el plano horizontal. El *Sodar* tiene una tasa de muestreo de 12 Hz, un rango de 2 metros, y una precisión de 5-10

centímetros, dependiendo en gran medida de la contaminación acústica y de las superficies reflectantes. Finalmente, la IMU ofrece la información de las aceleraciones, las tasas de cambio y la orientación a 76 Hz. Funciona realmente bien en ambientes con bajo nivel de vibraciones, perdiendo precisión según aumenta la amplitud de las vibraciones.

La estación de tierra es un ordenador portátil con Microsoft Windows XP y Labview 7, el cual se usa porque facilita el diseño de una GUI para la depuración, monitorización y control remoto de los vehículos. El vuelo manual se realiza mediante un joystick estándar conectado a la estación de tierra. El lazo de control se realiza usando un controlador PD.

En la literatura, por tanto, encontramos varios métodos para el control de UAVs. Se emplean realimentaciones de control robusto basados en técnicas H_{∞} [Chen, 2003], controladores tipo PID: [Bouabdallah *et al*, 2004] y [Pounds *et al*, 2002b] , PD [Hoffmann *et al*, 2004] y PI [Hanford, 2005]. Otros estudios de la universidad de Pensilvania utilizan controles basados en visión, [Altuğ, 2005]. En el capítulo 3 del presente trabajo se mostrarán dos estrategias de control diferentes: un LQR y control por modos deslizantes.

Con el fin de establecer estas leyes de control es necesario conocer el estado del sistema. Los parámetros necesarios para el control deberán ser medidos para implementar la realimentación. Con este objetivo se emplean los sensores, en los estudios que hemos visto todos los sistemas utilizan una unidad de medida inercial para el control de navegación del sistema. Esta unidad se compone de varios tipos de sensores: giróscopos que miden la tasa de cambio de la rotación alrededor de un eje determinado, acelerómetros que se usan para medir las aceleraciones lineales del sistema y pueden incluir brújula o magnetómetros con los que se puede detectar el campo magnético de la tierra para calcular la dirección y altitud.

Para determinar la posición del cuatrimotor en el exterior algunos de los estudios que se muestran emplean un GPS diferencial [Hanford, 2005] , [Brian, 2004] y [Hoffmann *et al*, 2004]. Este sistema cuando se utiliza en cooperación con la navegación inercial hace que el sistema sea más fiable. La sensorización del cuatrimotor construido para el proyecto de investigación en el que se encuadra este trabajo de master es una unidad de medida inercial con un giroscopio y un acelerómetro de tres ejes cada uno..

2. MODELADO Y ANÁLISIS DE LOS MOTORES

Con el fin de tener un modelo que se ajuste a la realidad lo mejor posible para el desarrollo de los controladores, en primer lugar se realiza un modelado de los motores. Para esta tarea se ha empleado un sistema que hemos llamado *Towercopter* del cual se pueden ver unas imágenes a continuación, Figura 10.

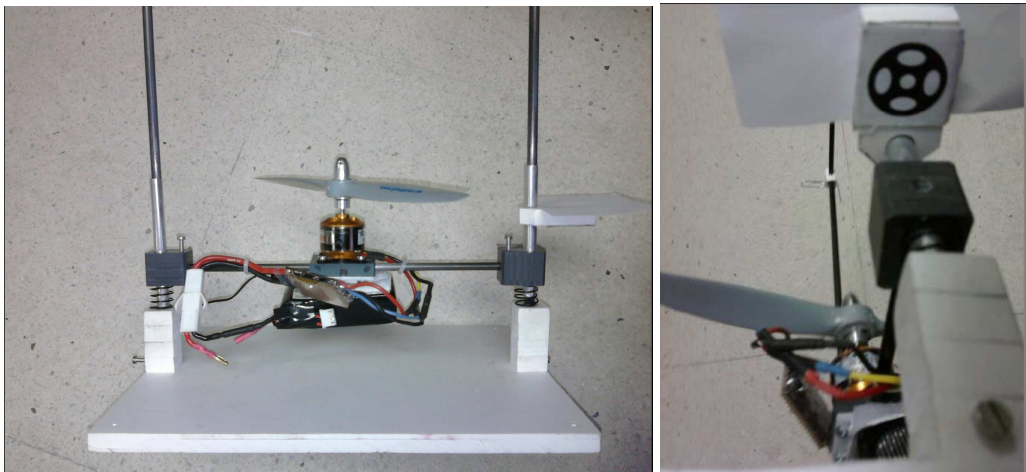


Figura 10. Vista frontal de *Towercopter* y vista desde abajo, esto es lo que la cámara ve.

El sistema se compone de una parte móvil que puede ascender siempre sujeto por unas guías verticales, que anulan cualquier otro grado de libertad que el sistema pudiera presentar, y una cámara nos sirve de sensor de altura con una original forma de determinación del estado.

Obtendremos la función de transferencia del sistema midiendo como datos de entrada, la altura de referencia que se le pide que alcance el sistema y como salida, las diferentes alturas por las que pasa el *Towercopter* a lo largo del tiempo.

2.1. Descripción del sistema

La parte principal del dispositivo consiste en un motor 1000 KV sin escobillas al que, directamente, se le acopla una hélice de 8 pulgadas de diámetro (20 centímetros). El valor 1000 KV significa que el motor es capaz dar 1000 rotaciones por minuto por cada voltio aplicado. El sistema se alimenta con una batería de polímero de Litio de alta descarga, con los beneficios que ello conlleva referentes a la potencia entregada y a un mayor tiempo de funcionamiento. Conectado al motor se encuentra un variador electrónico de velocidad (más comúnmente conocido como ESC, del inglés, *Electronic Speed Controller*) que puede suministrar hasta 30 Amperios al motor si este los demanda. Por su parte, el ESC está conectado a un pequeño dispositivo cuya función es comunicar y traducir las ordenes que el ordenador manda al ESC. El variador, necesita una señal de tipo PWM (*Pulse Position Modulation*) para adaptar, según dicha señal, la velocidad a la que girará el conjunto rotor.

La parte móvil de *Towercopter* incluye, además del motor, la bancada que lo sujeta y el variador de velocidad, dos brazos que salen lateral y opuestamente al rotor. Estos brazos están realizados en fibra de carbono, que da robustez y ligereza al sistema. Los brazos terminan con unas piezas de plástico, que a su vez llevan insertadas verticalmente a la línea que formaban los brazos unos tubos de aluminio que encajan a su vez en dos barras verticales, también de fibra de carbono. Así, los tubos de aluminio pueden desplazarse en el sentido de las barras verticales de carbono, llevando con ellos al resto de la plataforma móvil. La funcionalidad de estos tubos de aluminio es el que la parte ascendente no forme en ningún momento un ángulo con respecto al plano horizontal excesivo, lo que haría que el conjunto quedara bloqueado en su posición y anulara toda posibilidad de visualizar algo práctico.

2.1.1. Dinámica del sistema

La estructura de *Towercopter* presenta un único grado de libertad debido a que las barras laterales hacen que la plataforma móvil sólo pueda desplazarse en sentido vertical. Éstas introducen un rozamiento en el sistema que se opondrá siempre al sentido de movimiento. Por su parte, el cable que cuelga de la plataforma móvil presenta una masa mayor cuanto mayor es la altura a la que se encuentra ésta, hecho que tendremos de tener en cuenta a la hora de modelar el sistema. Y por supuesto estará presente la gravedad.

Empleando la segunda ecuación de la dinámica de Newton y prescindiendo de la notación vectorial debido a que el sistema se mueve sólo en una dirección la ecuación queda de la forma de (2.1)

$$M \ddot{x} = -Mg \pm F_{\text{rozamiento}} + F_{\text{empuje}} - F_{\text{cable}} \quad (2.1)$$

A continuación se detalla la forma de las fuerzas. La fuerza de rozamiento, depende directamente de la velocidad, ecuación (2.2). Cuanto más rápido se desplaza la plataforma a lo largo de las guías, menor resistencia al desplazamiento presentan éstas. Además, como hemos tomado

como negativa la fuerza de la gravedad y positivo el empuje, y sabemos que el rozamiento es opuesto al movimiento, debemos incluir el signo del desplazamiento de la plataforma.

$$F_{\text{rozamiento}} = -\frac{\text{signo}(\dot{x}) \cdot \mu_1}{1 + \mu_2 \dot{x}} \quad (2.2)$$

donde μ_1 y μ_2 son constantes determinadas experimentalmente.

La fuerza ejercida por el cable será mayor cuanto más cable esté colgando, lo que depende directamente de la altura a la que se encuentre la parte móvil. Si sabemos la masa del cable por cada centímetro de su longitud, tendremos la masa que cuelga de la plataforma. Con esto se escribe la ecuación (2.3)

$$F_{\text{cable}} = K \cdot x \quad (2.3)$$

donde K tiene dimensión de [gramos/centímetro (de cable)] y x representa la altura (espacio) a la que nos encontramos.

Combinando toda la información, obtenemos finalmente la ecuación de la dinámica de *Towercopter*, ecuación (2.4).

$$M \ddot{x} = -Mg - \frac{\text{signo}(\dot{x}) \cdot \mu_1}{1 + \mu_2 \dot{x}} + F_{\text{empuje}} - K \cdot x \quad (2.4)$$

2.1.2. Sensor de altura

A lo hora de controlar cualquier tipo de sistema, el ingeniero que lo desarrolla se ha de plantear entre otras cosas, el tipo de sensor a utilizar para tomar medida de las variables que considera importantes en el sistema. Existen muchos tipos de sensores, pero casi todos tienen algo en común, debemos tener conocimientos de electrónica para implantarlos en la planta y manejarlos, antes de poder medir con ellos.

Towercopter hace uso de otra estrategia. Aprovechando las oportunidades de análisis de imágenes que da MATLAB y lo barata que puede llegar a ser una cámara web, se calcula la altura a la que se encuentra la plataforma móvil mediante el uso de un algoritmo propio de visión artificial.

En esencia, se hace lectura del área de una determinada figura situada en un lateral externo de uno de los brazos que mantienen al rotor dentro de sus límites. Cuanto más distante es un objeto, más pequeño se nos presenta. Debemos tener en cuenta la pequeñez de la lente de la cámara, así como la curvatura de la misma. Estos factores, especialmente el segundo, hacen que la percepción del medio no sea lineal, y los objetos se vean afectados por aberraciones ópticas.

Las aberraciones esféricas son producidas por el hecho de que los puntos focales de rayos luminosos alejados del eje óptico (que pasa por el punto central) de una lente esférica son diferentes de los puntos focales de los rayos (de la misma longitud de onda) que pasan cerca del centro. Los rayos cercanos a la mitad del lente forman la imagen mas lejos de la lente que los rayos que la penetran por los bordes. En consecuencia, no hay una sola longitud focal para una lente. Como

ejemplo común podemos citar el astigmatismo que es una aberración que ocurre para objetos no localizados sobre el eje óptico del ojo.

Por lo tanto, previamente antes de usar la cámara, ésta ha de ser calibrada para obtener la expresión matemática que nos devuelva la distancia del objeto en función del área medida. De hecho, la toma de datos experimentales nos muestra que el área medida de la figura y la distancia a la que se mide sigue una relación de tipo potencial representada en la ecuación (2.5).

$$\text{Distancia} = K_1 \cdot \text{area}^{K_2} + K_3 \quad (2.5)$$

donde K_1 , K_2 y K_3 son constantes a determinar a partir de los datos experimentales. Así pues, nuestro algoritmo hace uso de una expresión así para calcular la distancia a la que se eleva la plataforma sobre el suelo.

La forma de la imagen que la cámara detecta tiene ciertas características que se describen a continuación, Figura 11.

Se puede observar como existen cinco regiones en el interior del disco. Existe una propiedad topológica de las figuras llamada *Número de Euler*, que es el número de regiones conexas del objeto (una en nuestro caso, la parte negra) menos el número de huecos que contiene (cinco según se cuenta). Así, para esta figura el número de Euler es -4. Esta particularidad la hace fácilmente distinguible del entorno, así si el algoritmo pierde de vista la figura le resulta sencillo volver a encontrarla para su análisis.

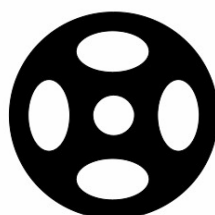


Figura 11. Imagen para la detección de la altura

2.2. Cálculo de la función de transferencia

2.2.1. Toma de datos

Para encontrar la función de transferencia se enviaron a *Towercopter* escalones de diferentes alturas, es decir, diferentes escalones de referencia, y capturar las diferentes alturas por las que pasa a lo largo del tiempo el sistema antes de estabilizarse en la altura deseada (ver Figura 12)

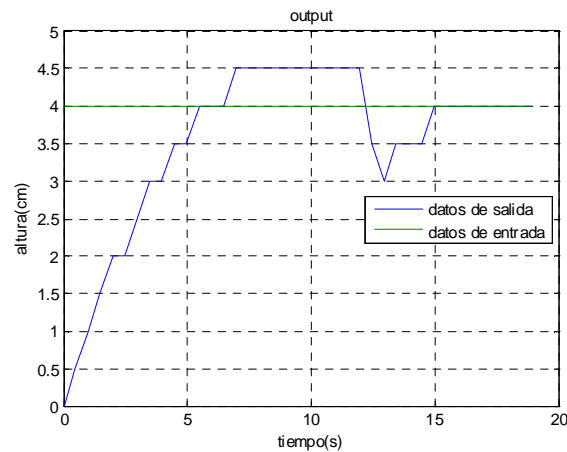


Figura 12. Representación de los datos experimentales

2.2.2. Función de transferencia

Vamos a utilizar como método para obtener la función de transferencia del sistema la herramienta de MATLAB para la identificación de modelos. Esta herramienta es muy simple de manejar, a continuación vamos a resumir los pasos principales que seguimos para obtener la función del sistema.

En primer lugar se importan los datos en la aplicación: la entrada, la salida, el tiempo y el periodo de muestreo. Toda esta información se saca de las pruebas experimentales: como datos de entrada introduciremos el valor de referencia, y datos de salida los valores de las alturas por las que pasa el sistema a lo largo del tiempo. Una vez tenemos importados los datos, elegiremos como modelo un sistema de primer orden, con una ganancia K y una constante de tiempo T , es decir, $\frac{K}{Ts+1}$.

Pocos segundos después tendremos los parámetros de nuestro sistema.

De las series de datos estudiadas se ha sacado un valor nominal para los parámetros que se ha tomado como función de transferencia del conjunto. A continuación podemos ver (Figura 13) la respuesta del sistema frente al tiempo, tanto para el caso real como para la respuesta obtenida por el modelo con los siguientes parámetros: $K = 1.0143$; $T_p = 1.7792$.

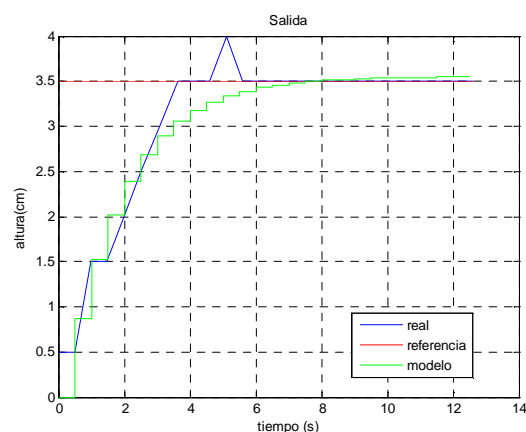


Figura 13. Respuesta del sistema real y modelado mediante ident

2.3. Verificación de la función de transferencia

En este apartado se muestra la Figura 14 con la comparación entre los datos experimentales tomados del *towercopter* y la comparación con el modelo que hemos obtenido mediante el sistema de identificación

Podemos ver en la Figura 14 como la línea azul representa los datos tomados experimentalmente, la línea verde se corresponde con la salida de la función de transferencia a una entrada dada por la referencia, línea roja en la figura. La salida de la función de transferencia es sobreamortiguada, pero tanto el tiempo de asentamiento como el de subida son similares a los de los datos experimentales. El modelo tiene un 5% de error estacionario. El modelo obtenido cumple con las características deseadas.

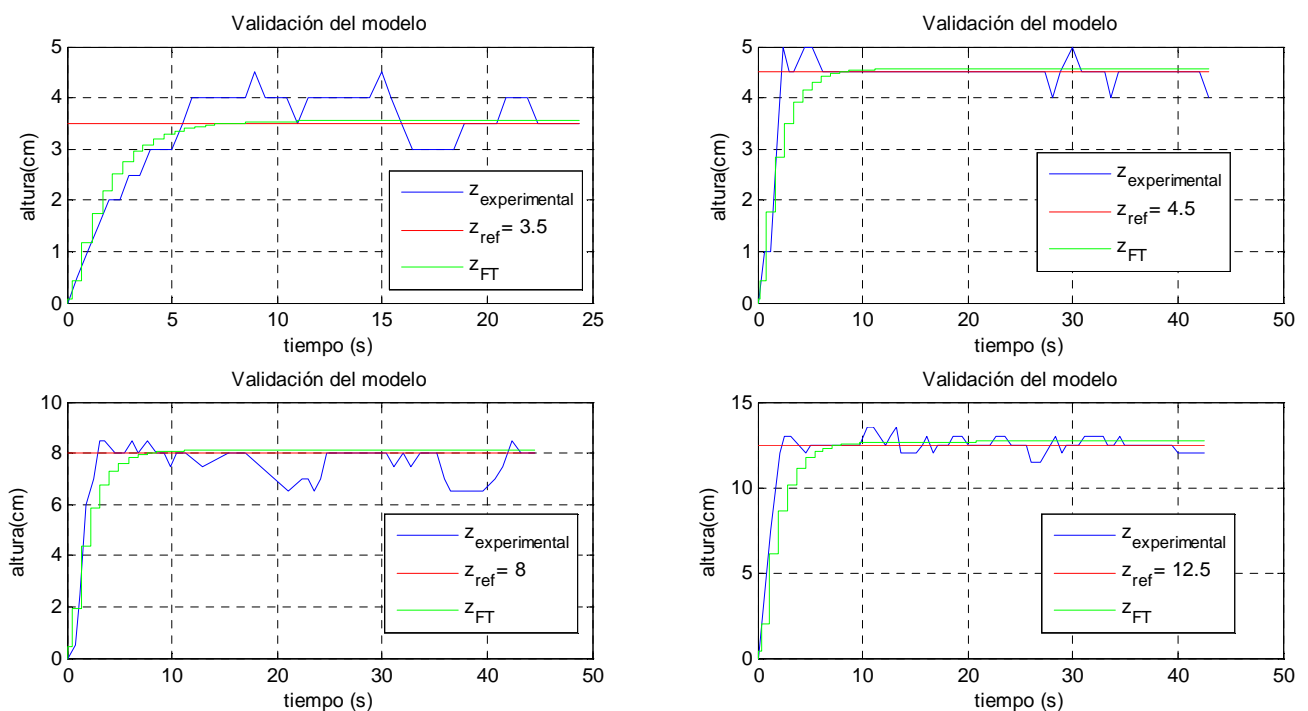


Figura 14. Validación del modelo

2.4. Comparación del sistema físico con el modelo extraído

A continuación vamos a comparar los resultados que se obtienen de simular la ecuación dinámica con la función de transferencia obtenida a partir de los datos experimentales. Emplearemos el diagrama SIMULINK que se muestra en la Figura 15.

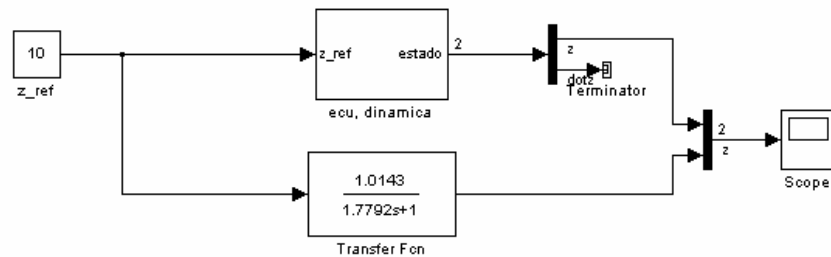


Figura 15. Diagrama SIMULINK para la comparación del modelo físico con la función de transferencia experimental

A continuación veremos cómo se ha implementado la ecuación dinámica en el subsistema de la Figura 16. Tenemos una función embebida de MATLAB que contiene la ecuación dinámica, tiene como salidas la velocidad y la aceleración del sistema. El bloque de limitación en altura es una saturación para la altura ya que el sistema no puede ascender a más de 25 cm.

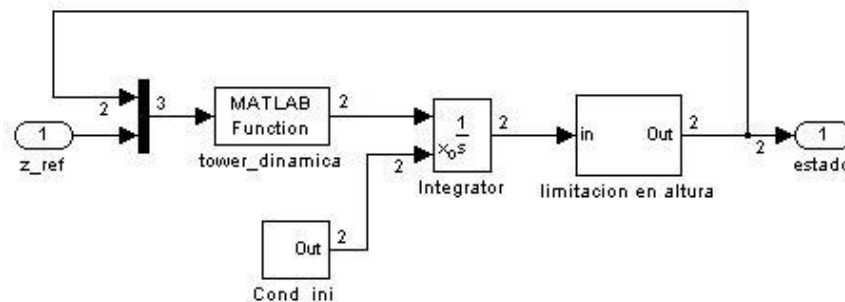


Figura 16. Subsistema Ecuación dinámica

Con este sistema hemos obtenido unas gráficas para diferentes valores de la altura de referencia para analizar las diferencias (Figura 17). La respuesta es diferente, porque la ecuación dinámica es un sistema de segundo orden mientras que la función de transferencia por la que estamos aproximando es de primer orden.

Observamos que el tiempo de subida es más rápido para la función de transferencia, pero el sistema tiene un tiempo de asentamiento similar para el sistema obtenido mediante la ecuación dinámica y el error es similar también en ambos casos. Por lo que concluimos que el modelo será válido pues se ajusta bien a los valores experimentales, como hemos demostrado en el apartado anterior, y obtenemos un buen ajuste a la ecuación física teniendo en cuenta que las respuestas se diferencian en el transitorio al ser el modelo físico y el experimental de distinto orden, pero en el estacionario se ajustan ambas perfectamente.

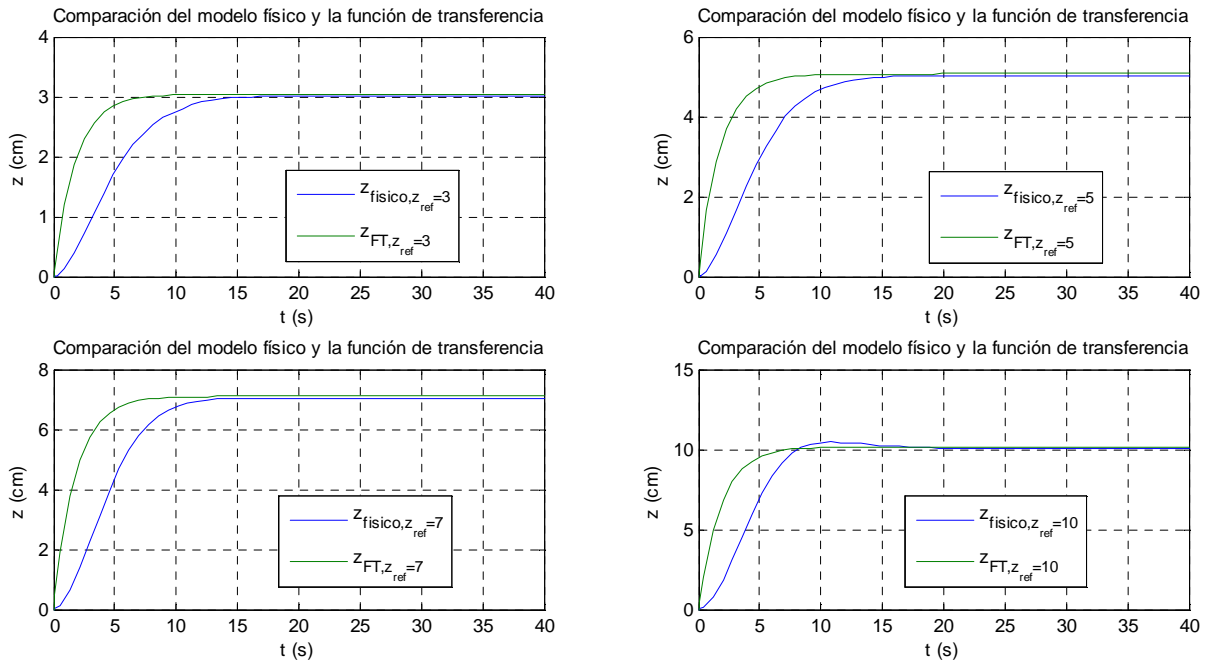


Figura 17. Salidas del diagrama SIMULINK

3. CONTROL DEL CUATRIMOTOR

El presente Capítulo se divide en cuatro grandes apartados. En el primero se hará una simplificación del modelo completo que se muestra en el *Apéndice A*, para comenzar a diseñar controladores de estabilización, en el segundo se realizará una linealización por realimentación de estado del modelo y a partir de ella obtendremos un controlador LQR y en el tercero se diseñará un controlador por modos deslizantes. El último apartado recoge el control en posición a partir del control de estabilización por modos deslizantes.

3.1. Simplificación del modelo

Para obtener un modelo con el que podamos empezar a trabajar haremos unas simplificaciones sobre la dinámica expuesta en el *Apéndice A*. Supondremos que el cuatrimotor está estabilizado a una cierta altura y además se tendrán en cuenta pequeñas variaciones en los ángulos entorno a esta posición de equilibrio. Centraremos nuestra atención en el control de los tres ángulos del sistema: alabeo, cabeceo y guiñada. De esta manera diseñaremos, en primer lugar, un control de estabilización para el sistema. Las ecuaciones dinámicas de nuestro sistema quedan resumidas en la ecuación (3.1)

$$\begin{bmatrix} J_y \ddot{\theta} + (J_z - J_x) \dot{\psi} \dot{\phi} \\ J_x \ddot{\phi} + (J_y - J_z) \dot{\psi} \dot{\theta} \\ J_y \ddot{\psi} + (J_x - J_y) \dot{\theta} \dot{\phi} \end{bmatrix} = \begin{bmatrix} (F_4 - F_2) d \\ (F_1 - F_3) d \\ \Omega_3 + \Omega_1 - \Omega_2 - \Omega_4 \end{bmatrix} = \begin{bmatrix} (V_4 - V_2) K_f l \\ (V_1 - V_3) K_f l \\ K_t (V_3 + V_1 - V_2 - V_4) \end{bmatrix} \quad (3.1)$$

Donde los momentos de inercia son: J_x, J_y, J_z (ecuación (0.14)); l es la distancia del centro del cuatrimotor al centro de las aspas. K_f y K_t son las constantes de fuerza que convierten la tensión aplicada a los motores en fuerzas desarrollada por las aspas para los motores

Tenemos cuatro señales de control para tres ángulos por lo que introducimos unas señales de control ficticias que definimos en la ecuación (3.2)

$$\begin{aligned}
V_1 &= \frac{\bar{u}}{4} - \frac{u_\phi}{2} + \frac{u_\psi}{4} \\
V_2 &= \frac{\bar{u}}{4} - \frac{u_\theta}{2} - \frac{u_\psi}{4} \\
V_3 &= \frac{\bar{u}}{4} + \frac{u_\phi}{2} + \frac{u_\psi}{4} \\
V_4 &= \frac{\bar{u}}{4} + \frac{u_\theta}{2} - \frac{u_\psi}{4}
\end{aligned} \tag{3.2}$$

De modo que las señales de control quedan de la forma que muestra (3.3)

$$\begin{aligned}
V_4 - V_2 &= u_\phi \\
V_1 - V_3 &= u_\theta \\
V_3 + V_1 - V_2 - V_4 &= u_\psi
\end{aligned} \tag{3.3}$$

El ángulo de cabeceo lo controlamos con u_ϕ , el ángulo de alabeo con u_θ y la guiñada con u_ψ . El término $\bar{u}=V_1+V_2+V_3+V_4$ indica el empuje que realiza cada motor con el sistema estacionario en una posición de equilibrio, y sería responsable de mantener girando al cuatrimotor a una altura equilibrando la fuerza de la gravedad; su variación permitiría subir o bajar al cuatrimotor. Tenemos ahora el sistema reducido a la forma de la ecuación (3.4)

$$\begin{bmatrix} \ddot{\theta} \\ \ddot{\phi} \\ \ddot{\psi} \end{bmatrix} = \begin{bmatrix} \frac{(J_z - J_x)}{J_y} \dot{\psi} \dot{\phi} \\ \frac{(J_y - J_z)}{J_x} \dot{\psi} \dot{\theta} \\ \frac{(J_x - J_y)}{J_z} \dot{\theta} \dot{\phi} \end{bmatrix} + \begin{bmatrix} \frac{K_f I}{J_y} u_\theta \\ \frac{K_f I}{J_x} u_\phi \\ \frac{K_t}{J_z} u_\psi \end{bmatrix} \tag{3.4}$$

Teniendo como variables de estado: $x = [\theta \ \phi \ \psi \ \dot{\theta} \ \dot{\phi} \ \dot{\psi}]^T$ y la salida es $y = [\theta \ \phi \ \psi]^T$ podemos reescribir el sistema, ecuación (3.5)

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \\ \dot{x}_4 \\ \dot{x}_5 \\ \dot{x}_6 \end{bmatrix} = \begin{bmatrix} x_4 \\ x_5 \\ x_6 \\ a_1 x_5 x_6 \\ a_2 x_4 x_6 \\ a_3 x_4 x_5 \end{bmatrix} + \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ b_1 & 0 & 0 \\ 0 & b_2 & 0 \\ 0 & 0 & b_3 \end{bmatrix} \begin{bmatrix} u_\theta \\ u_\phi \\ u_\psi \end{bmatrix} \tag{3.5}$$

$$y = [x_1 \ x_2 \ x_3]^T$$

3.2. Control LQR

El controlador LQR es una estrategia de control empleada en sistemas lineales por lo que el primer paso necesario para utilizar este control será hacer una linealización por realimentación de estados y seguidamente cerraremos el lazo de control diseñando un LQR.

3.2.1. Linealización por realimentación de estados

A continuación para linealizar el sistema realizaremos una realimentación de estados. Los problemas de realimentación de estado se plantean de la siguiente manera (Figura 18):

Tengamos un sistema no lineal que se puede escribir como en la ecuación (3.6)

$$\begin{aligned}\dot{x} &= f(x) + g(x)u \\ y &= h(x)\end{aligned}\quad (3.6)$$

donde $f(x)$ y $g(x)$ son funciones que admiten órdenes de derivadas tan elevados como sea necesario, se dice que son funciones suaves. Es posible encontrar una ley de control u que transforme al sistema en uno lineal y controlable, de la forma (3.7)

$$u = \alpha(x) + \beta(x)v \quad (3.7)$$

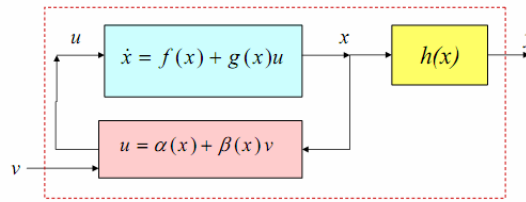


Figura 18. Problema planteado

La ecuación (3.8) muestra la evolución temporal de la salida.

$$\dot{y} = \frac{dh(x)}{dt} = \frac{\partial h(x)}{\partial x} \frac{dx}{dt} = \frac{\partial h(x)}{\partial x} f(x) + \frac{\partial h(x)}{\partial x} g(x)u = L_f h(x) + L_g h(x)u \quad (3.8)$$

donde $L_f h(x) = \frac{\partial h(x)}{\partial x} f(x)$ es la derivada de Lie de h con respecto a f o a lo largo de f . La derivada de Lie por tanto, es la velocidad de cambio de h a lo largo de las trayectorias del sistema $\dot{x} = f(x)$.

La elección de u para hacer el sistema lineal se toma de la forma (3.9)

$$\begin{aligned}u &= \frac{1}{L_g h(x)} (v - L_f h(x)) \\ \dot{y} &= v\end{aligned}\quad (3.9)$$

Si $L_g h(x) = 0$, entonces $\dot{y} = L_f h(x)$ es independiente de u . Si continuamos el cálculo de la segunda derivada, ecuación (3.10)

$$\ddot{y} = L_f^2 h(x) + L_g L_f h(x)u \quad (3.10)$$

Una vez más si $L_g L_f h(x) = 0$, entonces $\ddot{y} = L_f^2 h(x)$ es independiente de u y habría que recurrir a la siguiente derivada, si hacemos esto para el caso general, ecuación (3.11)

$$y^{(p)} = L_f^p h(x) + L_g L_f^{p-1} h(x)u \quad (3.11)$$

Definimos la entrada u , ecuación (3.12) para hacer el sistema lineal:

$$u = \frac{1}{L_g L_f^{\rho-1} h(x)} (v - L_f^{\rho} h(x)) \quad (3.12)$$

$$y^{\rho} = v$$

donde ρ es el grado relativo del sistema que de acuerdo con la siguiente definición es:

Definición: El sistema no lineal definido por la ecuación (3.6) tiene grado relativo ρ , $1 \leq \rho \leq n$, en la región $D_0 \subset D$ si $L_g L_f^{i-1} h(x) = 0$, $i=1,2,\dots, \rho-1$; $L_g L_f^{\rho-1} h(x) \neq 0$ para todo $x \in D_0$. Todo esto es extensible para sistemas con múltiples entradas y salidas, como es el caso de nuestro sistema.

El grado relativo del sistema que estamos estudiando es 2, por lo tanto, derivaremos dos veces la salida, ecuación (3.13)

$$\ddot{y} = L_f^2 h(x) + L_g L_f h(x) u \quad (3.13)$$

Y para el sistema dado por las ecuaciones del apartado 3.1. *Simplificación del modelo* tenemos:

$$L_f h(x) = \frac{\partial h(x)}{\partial x} f(x) = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} x_4 \\ x_5 \\ x_6 \\ a_1 x_5 x_6 \\ a_2 x_4 x_6 \\ a_3 x_4 x_5 \end{bmatrix} = \begin{bmatrix} x_4 \\ x_5 \\ x_6 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

$$L_g h(x) = \frac{\partial h(x)}{\partial x} g(x) = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} 0 & 0 & 0 \\ b_1 & 0 & 0 \\ 0 & b_2 & 0 \\ 0 & 0 & b_3 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

$$L_f^2 h(x) = \frac{\partial}{\partial x} \left(\frac{\partial h(x)}{\partial x} f(x) \right) f(x) = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} x_4 \\ x_5 \\ x_6 \\ a_1 x_5 x_6 \\ a_2 x_4 x_6 \\ a_3 x_4 x_5 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ a_1 x_5 x_6 \\ a_2 x_4 x_6 \\ a_3 x_4 x_5 \end{bmatrix}$$

$$L_g L_f h(x) = \frac{\partial}{\partial x} \left(\frac{\partial h(x)}{\partial x} f(x) \right) g(x) = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} 0 & 0 & 0 \\ b_1 & 0 & 0 \\ 0 & b_2 & 0 \\ 0 & 0 & b_3 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ b_1 & 0 & 0 \\ 0 & b_2 & 0 \\ 0 & 0 & b_3 \end{bmatrix}$$

Definimos con esto la entrada u que hace el sistema lineal de la forma (3.14)

$$u = \frac{1}{L_g L_f h(x)} (v - L_f^2 h(x)) = \begin{bmatrix} b_1 & 0 & 0 \\ 0 & b_2 & 0 \\ 0 & 0 & b_3 \end{bmatrix}^{-1} \left(\begin{bmatrix} v_1 \\ v_2 \\ v_3 \end{bmatrix} - \begin{bmatrix} a_1 x_5 x_6 \\ a_2 x_4 x_6 \\ a_3 x_4 x_5 \end{bmatrix} \right) = \begin{bmatrix} 1/b_1 & 0 & 0 \\ 0 & 1/b_2 & 0 \\ 0 & 0 & 1/b_3 \end{bmatrix} \begin{bmatrix} v_1 - a_1 x_5 x_6 \\ v_2 - a_2 x_4 x_6 \\ v_3 - a_3 x_4 x_5 \end{bmatrix}$$

$$\begin{bmatrix} u_\theta \\ u_\varphi \\ u_\psi \end{bmatrix} = \begin{bmatrix} \frac{v_1 - a_1 X_5 X_6}{b_1} \\ \frac{v_2 - a_2 X_4 X_6}{b_2} \\ \frac{v_3 - a_3 X_4 X_5}{b_3} \end{bmatrix} \quad (3.14)$$

Con lo que conseguimos un sistema lineal: $\ddot{y} = v$ (Figura 19)

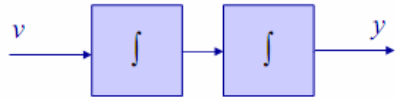


Figura 19. Sistema linealizado

Ahora que tenemos el sistema linealizado deberemos hacer un control sobre él. En primer lugar representaremos esta linealización por realimentación de estados en SIMULINK de MATLAB para ver cómo responde el sistema.

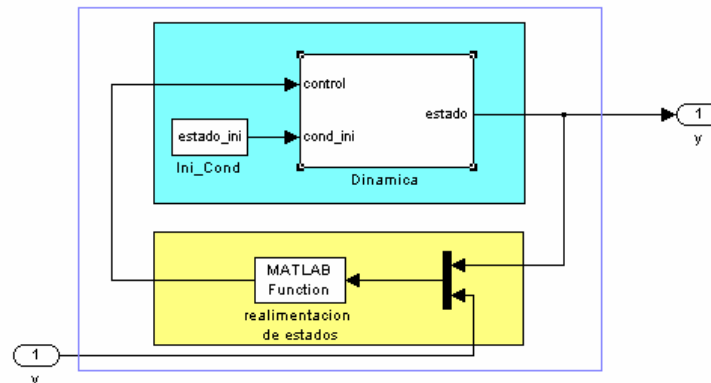


Figura 20. Dinámica del sistema y realimentación de estados

Vemos la correspondencia de bloques de la Figura 20 con lo que estábamos buscando, Figura 18. El bloque azul es la dinámica del sistema no lineal (este bloque representa un subsistema, se puede ver la composición en la Figura 21) El bloque amarillo representa la señal de control (3.7) que hace al sistema lineal. Tanto la dinámica como la realimentación de estados se ha implementado con funciones embebidas de MATLAB.

Este bloque de dinámica contiene una función embebida que tiene implementada las ecuaciones de la dinámica anteriormente planteadas, y toma como entradas de control las señales de control que genera la realimentación de estados y el estado anterior; como salida genera el nuevo estado.

La realimentación de estados fue implementada también como función embebida de MATLAB y tiene como entradas el estado y las señales de control v que definirá el controlador que le coloquemos al sistema. Como salida genera las u que hacen lineal la dinámica.

Si le introducimos unas señales v de referencia para ver como se comporta el sistema siendo v_1 una entrada escalón de 0.1 y v_2 y v_3 nulas la salida que esperamos es una respuesta en alabeo, y como es una constante aumentará el ángulo durante todo el tiempo que dure la señal. El cabeceo y la

guiñada no se verán afectados. Esto se ve en la Figura 22.a, mientras que en la Figura 22.b hemos introducido una señal de 0.1 tanto en v_1 como en v_2 manteniendo v_3 nula.

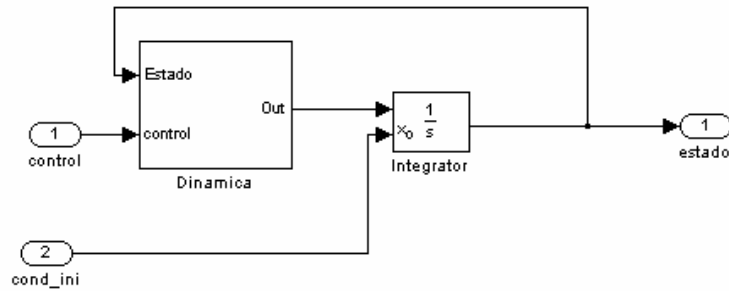


Figura 21. Subsistema dinámico

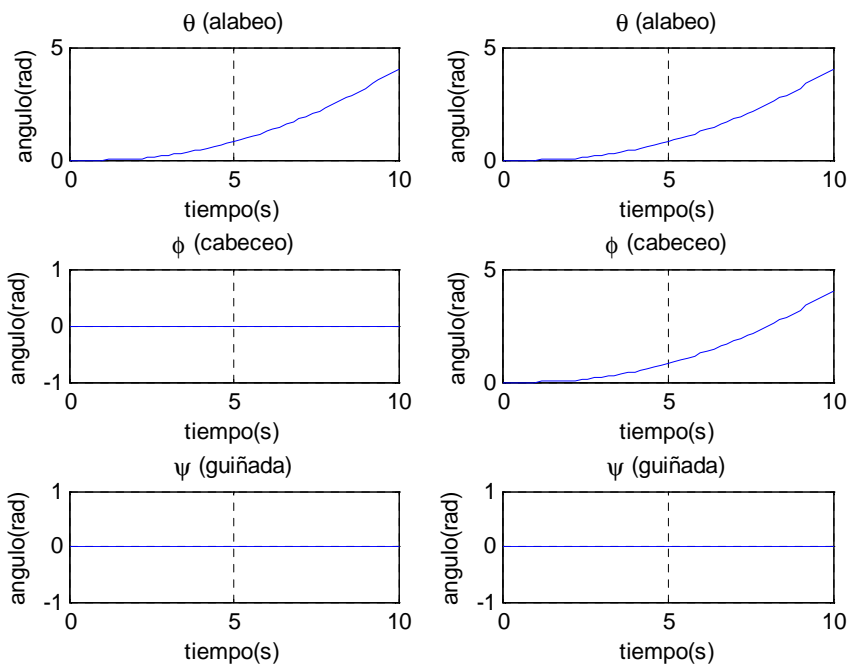


Figura 22. Respuestas del sistema

3.2.2. Controlador LQR

Para cerrar el lazo de realimentación emplearemos un controlador LQR. El problema de encontrar el controlador LQR óptimo busca hacer el siguiente criterio de la ecuación (3.15) lo más pequeño posible.

$$J_{LQR} := \int_0^{\infty} (x'Qx + u'Ru) dt \quad (3.15)$$

Donde Q es una matriz $n \times n$ y R es $m \times m$, ambas deben ser simétricas definidas positiva. Vemos que el término en x (los estados) corresponde a la energía de la salida controlada, mientras que el término cuadrático en u (señal de control) corresponde a la energía del control. Se busca que ambos sean mínimos, pero el decremento en la energía de la salida requerirá grandes señales de control, y una señal pequeña de control dejará grandes señales de salida.

Por lo tanto, la primera cuestión que se nos plantea es por donde empezar a buscar matrices Q y R, la primera elección podemos hacerla según la regla de Bryson, que dice:

Se seleccionan matrices Q y R diagonales con los valores de la ecuación (3.16)

$$Q_{ii} = \frac{1}{\text{máximo aceptable de } x_i^2} \quad i \in \{1,2,\dots,l\}$$

$$R_{jj} = \frac{1}{\text{máximo aceptable de } u_j^2} \quad j \in \{1,2,\dots,m\}$$
(3.16)

Aunque la regla de Bryson a veces produce buenos resultados, la mayoría de las veces es un punto de partida para un procedimiento de ensayo y error iterativo dirigido a la obtención de unas propiedades deseadas en el lazo cerrado.

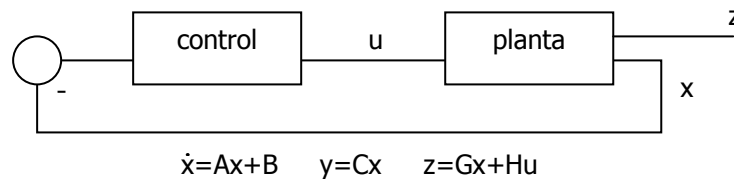


Figura 23. Regulador LQR

El controlador óptimo LQR para un problema como el de la Figura 23 tiene la forma de una matriz de ganancia, dado en la ecuación (3.17)

$$u = -Kx \tag{3.17}$$

donde K es una matriz $m \times n$ dada por la ecuación (3.18)

$$K = (H'QH + \rho R)^{-1} (B'P + H'QG) \tag{3.18}$$

Y P es la única solución definida positiva de la ecuación de Riccati, definida en (3.19)

$$A'P + PA + G'QG - (PB + G'QH)(H'QH + \rho R)^{-1} \tag{3.19}$$

MATLAB tiene implementada en su toolbox de control la función LQR que resuelve las ecuaciones que acabamos de ver.

$$[K, S, E] = \text{LQR}(A, B, Q, R)$$

Como argumentos de entrada introduciremos A y B que son las matrices que definen al sistema real, Q y R que son las matrices que buscamos que hagan J_{LQR} mínimo. Los argumentos de salida de la función son: K, la matriz de ganancias para el controlador; S, solución a la ecuación de Riccati y E que son los autovalores de $[A - BK]$.

Nuestro sistema tiene la forma: $\dot{x} = Ax + b$

$$A = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}; \quad b = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

Las matrices Q y R se han obtenido por diferentes pruebas en ensayo y error de los valores, aunque en la primera iteración se empleó la regla de Bryson. A continuación se muestran las matrices

Q y R que se obtuvieron en la última prueba con las que se consiguen las propiedades deseadas para el sistema.

$$R = \begin{bmatrix} 0.4 & 0 & 0 \\ 0 & 0.4 & 0 \\ 0 & 0 & 0.4 \end{bmatrix} ; \quad Q = \begin{bmatrix} 0.9549 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0.9549 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0.9549 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0.1000 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0.1000 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0.1000 \end{bmatrix}$$

Introduciendo estos datos en la función de MATLAB `lqr`, la matriz de ganancias que obtenemos es la siguiente:

$$K = \begin{bmatrix} 1.5451 & 0 & 0 & 1.8276 & 0 & 0 \\ 0 & 1.5451 & 0 & 0 & 1.8276 & 0 \\ 0 & 0 & 0.6308 & 0 & 0 & 1.2295 \end{bmatrix}$$

Implementamos este controlador en nuestro sistema como vemos en la Figura 24. El bloque morado que tenemos en este modelo SIMULINK es el que obtuvimos de la realimentación de estados, Figura 20.

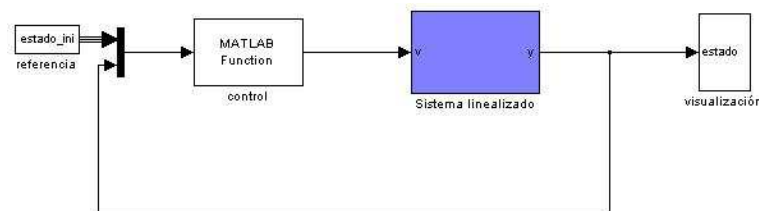


Figura 24. Sistema controlado mediante un regulador LQR

A continuación veremos una simulación con unos valores de referencia dados, Figura 25: $\theta = 0.1$ rad; $\phi = 0.17$ rad ; $\psi = 0.1$ rad

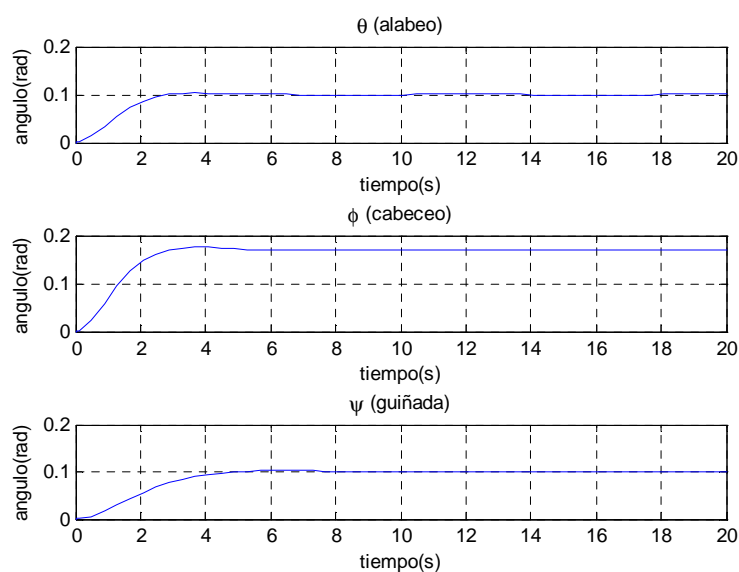


Figura 25. Respuestas del sistema controlado

3.3. Control por modos deslizantes

Para diseñar un control por modos deslizantes partimos de unas ecuaciones dinámicas que escribimos de la forma (3.20)

$$\begin{aligned}\dot{x}_1 &= x_2 \\ \dot{x}_2 &= h(x) + g(x)u\end{aligned}\quad (3.20)$$

donde h y g son funciones desconocidas y no lineales. El diseño del control tiene dos fases, en la primera buscamos una ley de control que establezca el sistema en el origen. Supongamos que nosotros podemos diseñar una ley de control que limite el movimiento del sistema a una superficie $s = \lambda x_1 + x_2 = 0$ (esto es idéntico a $s = \left(\frac{d}{dt} + \lambda\right)^n x_1$ con $n=1$). En este plano, el movimiento está gobernado por $\dot{x}_1 = -a_1 x_1$, y eligiendo $a_1 > 0$ garantizamos que x_1 tiende a cero cuando el tiempo tiende al infinito y el radio de convergencia está controlado por el valor de a_1 .

La ley de control equivalente toma la forma de la ecuación (3.21)

$$\begin{aligned}\dot{s} &= \lambda \dot{x}_1 + \dot{x}_2 = \lambda x_2 + h(x) + g(x)u_{eq} = 0 \\ u_{eq} &= -\frac{\lambda x_2 + h(x)}{g(x)}\end{aligned}\quad (3.21)$$

La segunda fase consiste en encontrar una ley de control que cambie, de tal forma que si se está fuera de la superficie $s=0$, entonces los vectores tangente a las trayectorias estado apunten hacia dicha superficie, haciéndolas de este modo tender a ellas. Para lo cual definiremos la función de Lyapunov, (3.22)

$$V = \frac{1}{2} s^2 \quad (3.22)$$

Esta función cumple (3.23)

$$\dot{V} = s\dot{s} \leq 0 \quad (3.23)$$

Para nuestro sistema tomaremos una de las estructuras más usadas [Levine, 1996] (3.24)

$$u_N = -k \cdot \text{sgn}(s) \quad (3.24)$$

La ley de control completa será la suma de (3.21) y (3.24), ecuación (3.25)

$$u(x,t) = u_{eq}(x,t) + u_N(x,t) \quad (3.25)$$

Aplicamos esto a las ecuaciones simplificadas del cuatrímotor, ecuación (3.4), para obtener la ley de control. Definimos las superficies ($s = \dot{x}_1 + \lambda x_1$) y sus derivadas como se expresa en (3.26)

$$\begin{aligned}s_\theta &= \dot{\theta} + \lambda_1 (\theta - \theta_d) \rightarrow \dot{s}_\theta = \ddot{\theta} + \lambda_1 \dot{\theta} \rightarrow \dot{s}_\theta = \frac{(J_z - J_x)}{J_y} \dot{\psi} \dot{\phi} + \frac{K_f I}{J_y} u_\theta + \lambda_1 \dot{\theta} \\ s_\phi &= \dot{\phi} + \lambda_1 (\phi - \phi_d) \rightarrow \dot{s}_\phi = \ddot{\phi} + \lambda_1 \dot{\phi} \rightarrow \dot{s}_\phi = \frac{(J_y - J_z)}{J_x} \dot{\psi} \dot{\theta} + \frac{K_f I}{J_x} u_\phi + \lambda_1 \dot{\phi} \\ s_\psi &= \dot{\psi} + \lambda_2 (\psi - \psi_d) \rightarrow \dot{s}_\psi = \ddot{\psi} + \lambda_2 \dot{\psi} \rightarrow \dot{s}_\psi = \frac{(J_x - J_y)}{J_z} \dot{\theta} \dot{\phi} + \frac{K_t}{J_z} u_\psi + \lambda_2 \dot{\psi}\end{aligned}\quad (3.26)$$

Haciendo $\dot{s} = 0$ obtenemos las u_{eq} de los tres controladores, ecuación (3.27)

$$\begin{aligned} u_{eq,\theta} &= -\frac{J_y}{K_f I} \left\{ \frac{(J_z - J_x)}{J_y} \dot{\psi} \dot{\phi} + \lambda_1 \dot{\theta} \right\} \\ u_{eq,\phi} &= -\frac{J_x}{K_f I} \left\{ \frac{(J_y - J_z)}{J_x} \dot{\psi} \dot{\theta} + \lambda_1 \dot{\phi} \right\} \\ u_{eq,\psi} &= -\frac{J_z}{K_t} \left\{ \frac{(J_x - J_y)}{J_z} \dot{\theta} \dot{\phi} + \lambda_2 \dot{\psi} \right\} \end{aligned} \quad (3.27)$$

Definiendo la parte u_N por: $u_N = -k \cdot \text{sgn}(s)$. Tenemos que los controles por modos deslizantes para cada ángulo que vemos en la ecuación (3.28)

$$\begin{aligned} u_\theta &= -\frac{J_y}{K_f I} \left\{ \frac{(J_z - J_x)}{J_y} \dot{\psi} \dot{\phi} + \lambda_1 \dot{\theta} \right\} - k_1 \text{sgn}(\dot{\theta} + \lambda_1 (\theta - \theta_d)) \\ u_\phi &= -\frac{J_x}{K_f I} \left\{ \frac{(J_y - J_z)}{J_x} \dot{\psi} \dot{\theta} + \lambda_1 \dot{\phi} \right\} - k_1 \text{sgn}(\dot{\phi} + \lambda_1 (\phi - \phi_d)) \\ u_\psi &= -\frac{J_z}{K_t} \left\{ \frac{(J_x - J_y)}{J_z} \dot{\theta} \dot{\phi} + \lambda_2 \dot{\psi} \right\} - k_2 \text{sgn}(\dot{\psi} + \lambda_2 (\psi - \psi_d)) \end{aligned} \quad (3.28)$$

Implementamos el nuevo controlador en nuestro modelo SIMULINK (Figura 26)

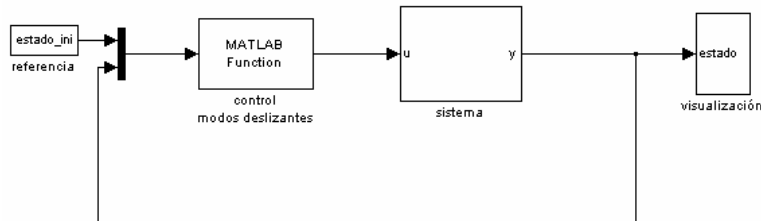


Figura 26. Sistema controlado mediante un control por modos deslizantes

El resultado de la simulación se puede observar en la Figura 27. las referencias usadas para dicha simulación son: $\theta = 0.1$ rad; $\phi = 0.17$ rad ; $\psi = 0.1$ rad

En la Figura 27.a. Se puede observar que controlamos el sistema y se llega con bastante rapidez a los ángulos de referencia marcados. La guiñada (ψ) es el ángulo que más tarda en estabilizarse en el valor deseado. En la Figura 27.b podemos observar que la respuesta del controlador está oscilando constantemente, este efecto se conoce como *chattering*.

En la Figura 28 se muestra cómo los retardos pueden causar el *chattering*. Representa a una trayectoria en la región $s > 0$ parte hacia la superficie $s = 0$ que actúa como un atractor. En primer lugar, llega en el punto a . En el control de deslizamiento ideal, debe comenzar la trayectoria de deslizamiento sobre la superficie desde el punto a . En realidad, habrá un retardo entre el momento en el cambio del signo de s y el tiempo en los cambios del control. Durante este período, la trayectoria invierte su dirección y se dirige nuevamente hacia la superficie $s = 0$. Una vez que la cruza la superficie atractora, y la repetición de este proceso crea el zig-zag (oscilación).

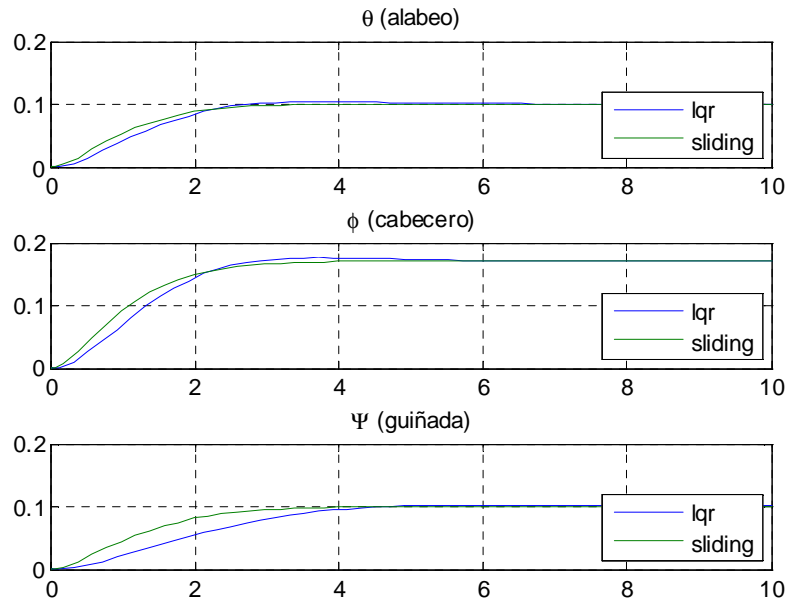


Figura 30. Comparación de los controladores en ángulos

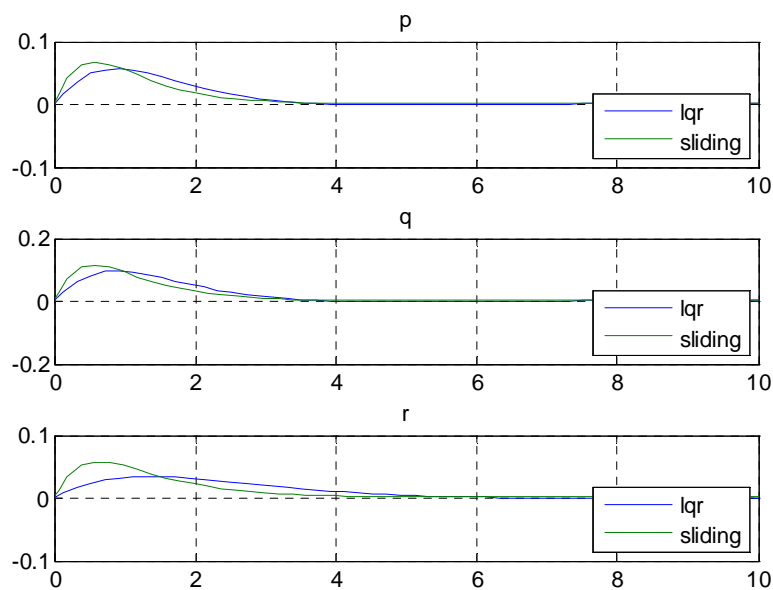


Figura 31. Comparación de los controladores en velocidad

3.5. Control en posición del cuatrimotor

A continuación, en la Figura 32 se representa el modelo SIMULINK del cuatrimotor completo. En este modelo no se tienen en cuenta las limitaciones propuestas es en el apartado *3.1.Simplificación del modelo* porque en este caso dejaremos que nuestro modelo tenga los 6 grados de libertad. Por lo que emplearemos para simular la dinámica del sistemas las ecuaciones (0.3), (0.5), (0.12) y (0.19).

Fijemos la atención en la Figura 32, tenemos tres bloques que se encargan de simular el comportamiento del cuatrimotor (bloques en amarillo en la figura) que son: la *S-function* en el modelo Dinámica_Cuatrimotor, el bloque de limitaciones y el que se encarga de simular la dinámica de los motores. Por otra parte tenemos el bloque de control, bloque verde, y el de referencias, en naranja en el modelo, y se han colocado unos bloques para representar el estado, las señales de control y las señales que salen de los motores, representados todos ellos en azul.

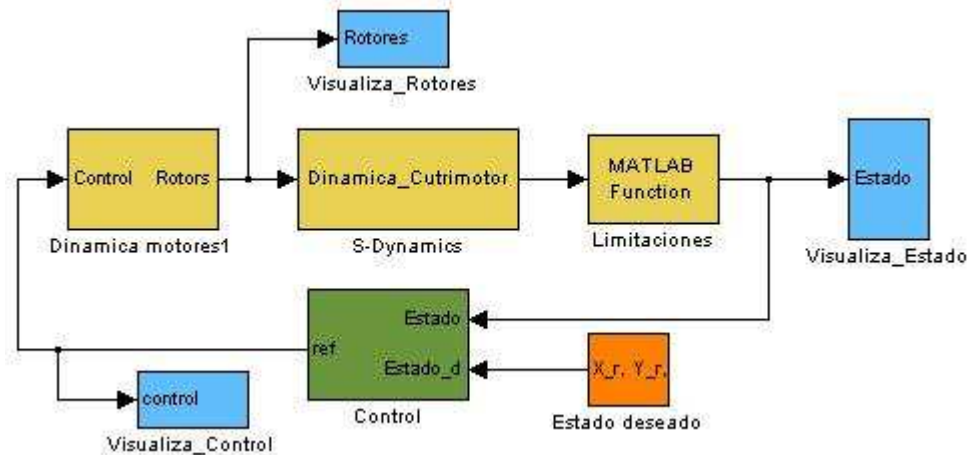


Figura 32. Modelo completo de cuatrimotor

Centremos nuestra atención en primer lugar en la dinámica del cuatrimotor. El bloque más importante de esta serie de bloques es la *S-function* de MATLAB que se encarga de coger las ecuaciones diferenciales del cuatrimotor, y con unas condiciones iniciales dadas las integra, y nos devuelve el estado y realimenta con este estado el bloque para volver a empezar. La dinámica se representa en la función de MATLAB *s_dina_fun.m*.

Para la simulación de la dinámica además tenemos dos bloques más, por un lado uno que se encarga de las limitaciones físicas, en esencia, se encarga de que el cuatrimotor no pueda estar a una altura por debajo de cero, en caso de que sea cero, es decir, halla llegado al suelo, las velocidades de los motores van a ser cero. Y limita los valores de los tres ángulos de Euler para que estén entre $[-\pi, \pi]$. El otro bloque se encarga de simular la dinámica de los motores y vemos la estructura interna de este subsistema en la Figura 33. La función embebida de MATLAB se encarga de transformar las señales de control en señales para los motores, mediante la ecuación (3.3), se han colocado unos retardos entre que se ha generado la señal de control y llega a los motores, así como entre la actuación de los motores y que el sistema de cuenta de este cambio. Para simular los motores se ha usado la función de transferencia que se obtuvo en el apartado 2.2.2. *Función de transferencia* de este proyecto. Finalmente esta señal se satura, pues los motores tienen unas revoluciones máximas a las que giraran.

La estructura del bloque de control se muestra en la Figura 34. Se ve como se divide en el control en posición y el control que se encarga de la estabilización del sistema. El control en posición, tiene como entradas el estado y las posiciones de referencia (x, y, z), este bloque genera los ángulos

de referencia que envía al bloque de estabilización. En este caso elegimos como control para la estabilización el modos deslizantes, como se explicó en el apartado 3.4.Comparación de los controles.

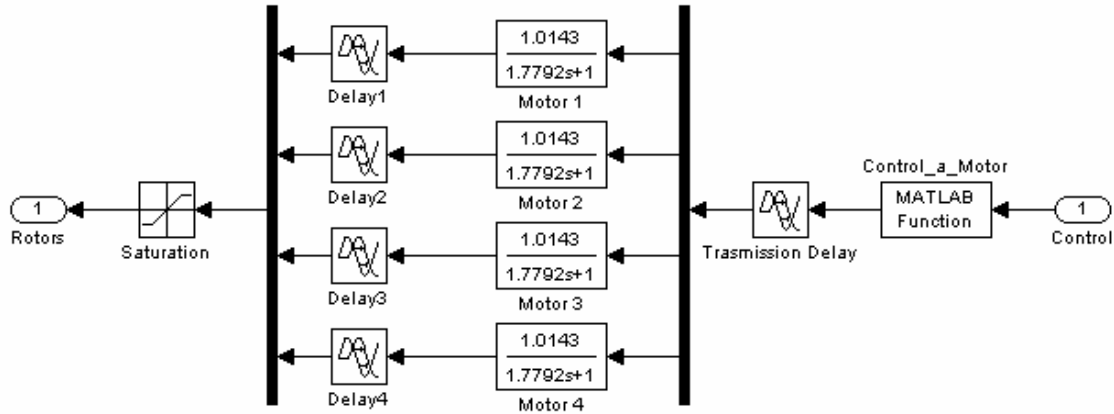


Figura 33. Subsistema dinámico de los motores

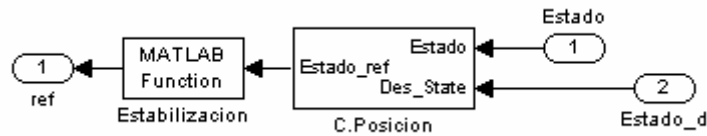


Figura 34. Subsistema control

La altura del sistema es una variable desacoplada, por lo que diseñaremos una ley de control de tipo modos deslizantes para ella. Definamos la superficie de deslizamiento en la ecuación (3.29)

$$s_z = \dot{z} + \lambda(z - z_d) \rightarrow \dot{s}_z = \ddot{z} + \lambda\dot{z} \rightarrow \dot{s}_z = QU - PV + \cos(\phi)\cos(\theta)g - \frac{u_z}{M_T} + \lambda\dot{z} \quad (3.29)$$

Haciendo $\dot{s} = 0$ obtenemos las u_{eq} que vemos en la ecuación (3.30)

$$u_{eq} = (QU - PV + \cos(\phi)\cos(\theta)g + \lambda\dot{z})M_T \quad (3.30)$$

Definiendo la parte u_N por: $u_N = -k \cdot \text{sgn}(s)$ y sustituyendo la función signo por una saturación para evitar el chattering como hicimos en el apartado 3.3.Control por modos deslizantes se define la ley de control para la altura de la forma que se ve en (3.31)

$$u_z = (QU - PV + \cos(\phi)\cos(\theta)g + \lambda\dot{z})M_T - k \cdot \text{sat}(\dot{z} + \lambda(z - z_d)) \quad (3.31)$$

Para la posición en (x, y) se ha empleado un control tipo PD que genera las señales de referencia para los ángulos.

A continuación se ejecutará el modelo SIMULINK de la Figura 32 para diferentes condiciones iniciales y condiciones finales a la que se desea que el sistema llegue. En la Figura 35 podemos ver la simulación para condiciones iniciales $\theta = 0.1$ rad; $\phi = 0.1$ rad ; $\psi = 0.1$ rad; $x = -2$ m; $y = -2$ m; $z = 0$ m; la posición a la que deseamos que el cuatrимotor llegue es $x = 0$ m; $y = 0$ m; $z = 4$ m;

Vemos que en la Figura 35 en menos de cinco segundos se ha alcanzado el estado que hemos dado como referencia al sistema. Una vez se ha alcanza la posición deseada los ángulos se anulan para que el sistema no se siga moviendo, porque recordemos que los ángulos de alabeo y cabeceo están acoplados a las posiciones x e y.

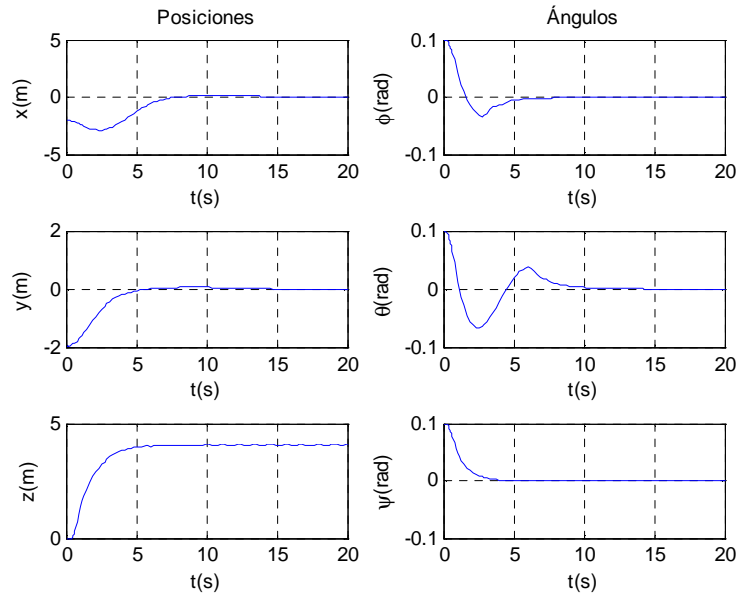


Figura 35. Simulación del modelo del control completo del cuatrимotor

Finalmente lanzaremos otra simulación, que podemos ver en la Figura 36, con condiciones iniciales $\theta=0$ rad; $\phi=0$ rad ; $\psi=0$ rad ; $x=0$ m; $y=0$ m; $z=4$ m; y cuya posición final a la que se desea que llegue es $x=-3$ m; $y=-3$ m; $z=1$ m;

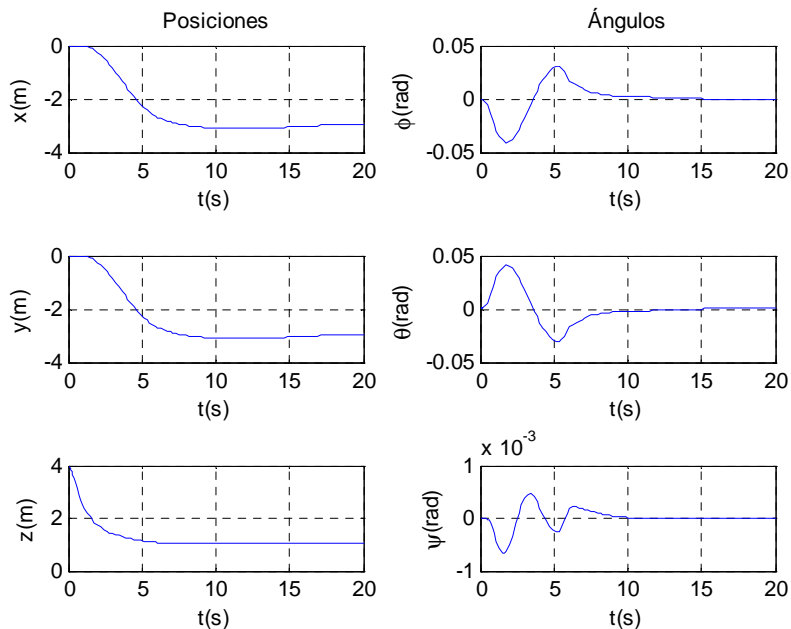


Figura 36. Simulación del modelo del control completo del cuatrимotor

A la vista de las gráficas podemos concluir que el sistema cumple con las consignas dadas en un tiempo aceptable. Una vez concluida la fase de simulaciones se pasa a la fase de pruebas reales de los controles de estabilización diseñados.

4. PRUEBAS REALES

Para las pruebas reales se han empleado una plataforma sobre la que se han realizado pruebas con los controladores expuestos en el Capítulo 3 del presente trabajo. El sistema ha sido adquirido a Quanser. [Quanser].

4.1. Torre 3D Hover de Quanser

Este modelo de cuatrimotor está fijo y sólo permite, por tanto, los movimientos angulares lo que permite controlar sus ángulos de cabeceo, alabeo y guiñada. La Figura 37 muestra el sistema. En esta plataforma vamos a aplicar los resultados de control que se han obtenido según la simplificación hecha en el apartado *3.1.Simplificación del modelo*.



Figura 37. 3D Hover de Quanser

En la Tabla 1 se resume la información mecánica de la estructura del sistema empleado en los modelos SIMULINK que se explicaran a continuación. Además en la Tabla 2 se muestran parámetro de las hélices. Finalmente en la Tabla 3 se muestran parámetro de los motores empleados en la plataforma.

Símbolo	Descripción	Valor
M	Masa total del cuerpo	2.85 Kg
m	Masa por motor	0.7125 Kg
l	Distancia del punto de giro al motor	0.1969 m
J_{ϕ}	Momento de inercia en el eje del alabeo	0.0552 Kg m ²
J_{θ}	Momento de inercia en el eje del cabeceo	0.0552 Kg m ²
J_{ψ}	Momento de inercia en el eje de la guiñada	0.1104 Kg m ²

Tabla 1. Parámetros de la estructura mecánica del sistema

Símbolo	Descripción	Valor
	Dimensiones	8 x 6 pulgadas
K_f	Constante de fuerza	0.5940 N/V
K_t	Constante de momento	0.0180 Nm/V
T_{aup}	Constante de tiempo	0.15 s

Tabla 2. Parámetros de las hélices

Parámetro	Valor
Resistencia de la armadura	0.83 Ω
Constante de momento	0.0182 Nm/A
Tensión nominal	12V

Tabla 3. Parámetros de los motores Pittman 9234S004

A continuación vamos a mostrar cómo se han implementado estos modelos en SIMULINK con los bloques para la adquisición de datos de las tarjetas que lleva incorporadas la torre de Quanser para la lectura de sus encoders. Finalmente, para cada controlador se han extraído unos datos del aparato en funcionamiento.

4.1.1. Realimentación de estados y controlador LQR

En este apartado vamos a implementar el controlador LQR diseñado tras la realimentación de estados que se hizo en el modelo para la linealización del sistema. El modelo SIMULINK completo se muestra en la Figura 38. En primer lugar, fijaremos nuestra atención en los bloques de entrada y salida que proporciona Quanser para la comunicación con la tarjeta de adquisición de datos, Figura 39 y Figura 40.

En la Figura 39 podemos ver el subsistema que contiene los encoders y unos estimadores de la velocidad, las salidas de este subsistema son los estados para nuestro sistema. Los tres ángulos y las tasas de cambio de estos ángulos.

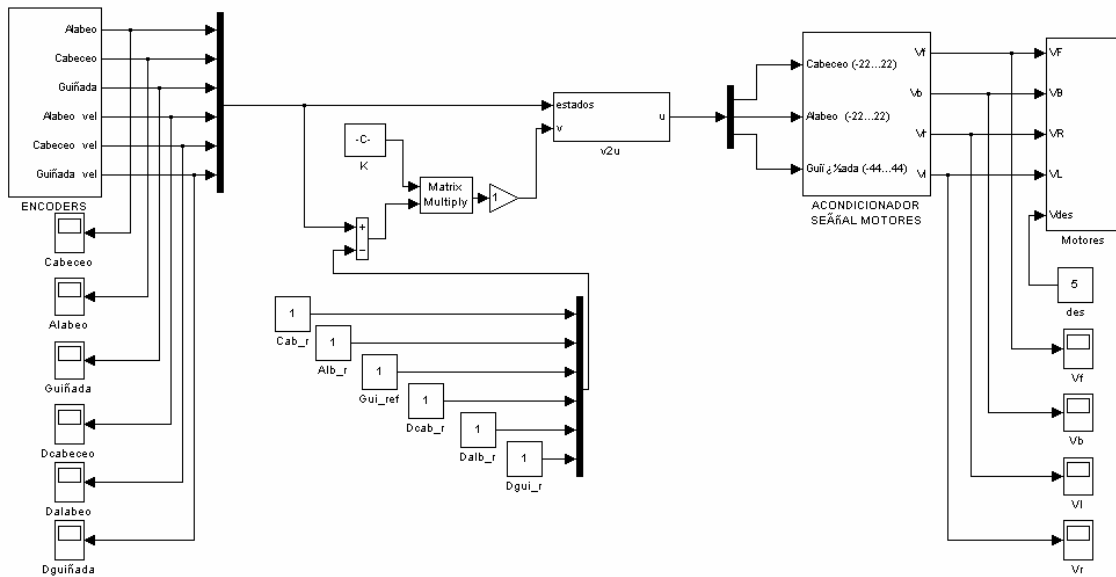


Figura 38. Controlador LQR para el sistema 3D Hover de Quanser

La Figura 40 muestra el subsistema de salida que manda al sistema las señales para los motores. Toma como entradas las señales que saca el bloque que acondiciona las señales de control a señales para los motores mediante la formula (5.1), y la saca por los motores.

$$V_1 = \frac{\bar{u}}{4} - \frac{u_\phi}{2} + \frac{u_\psi}{4}; V_2 = \frac{\bar{u}}{4} - \frac{u_\theta}{2} - \frac{u_\psi}{4}; V_3 = \frac{\bar{u}}{4} + \frac{u_\phi}{2} + \frac{u_\psi}{4}; V_4 = \frac{\bar{u}}{4} + \frac{u_\theta}{2} - \frac{u_\psi}{4} \quad (5.1)$$

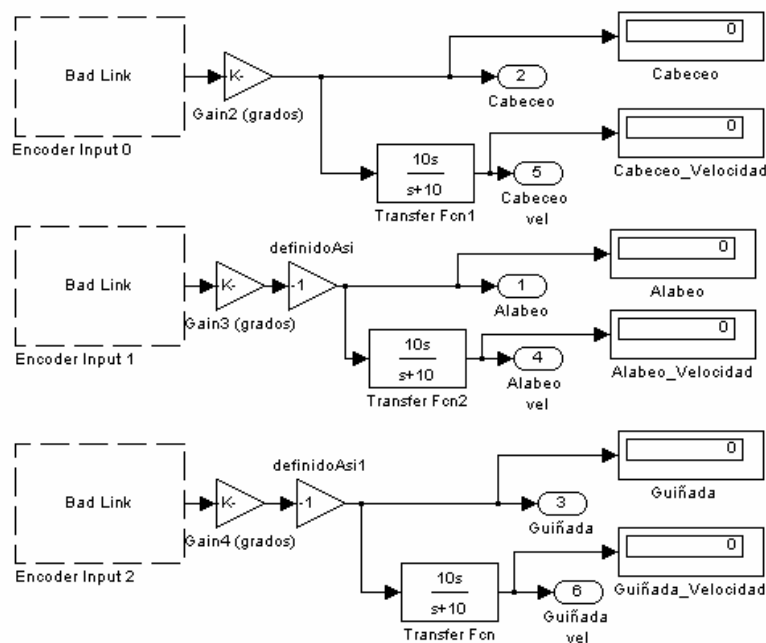


Figura 39. Subsistema encoders

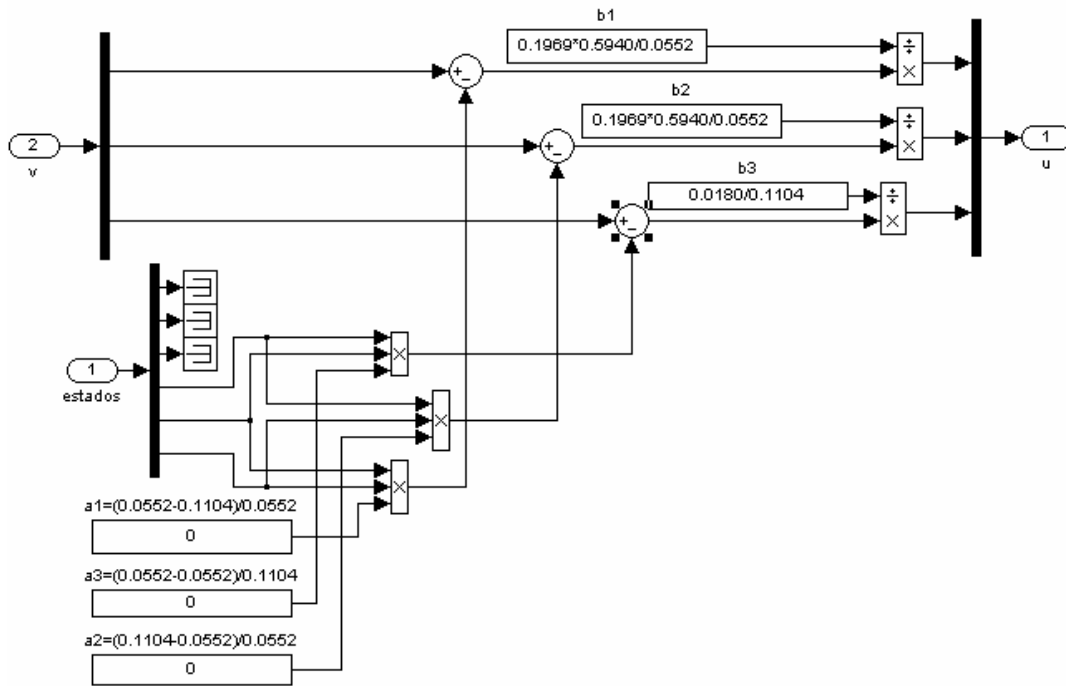


Figura 42. Subsistema para la realimentación de estados

El sistema real no sigue bien la dinámica propuesta por lo que para que no se desestabilizara en las pruebas reales se han anulado las constantes a 's, consiguiéndose así la estabilización del sistema y los resultados que mostraremos a continuación.

Para obtener las siguientes gráficas se ha desestabilizado el sistema en cada uno de los ejes y se ha esperado a que el sistema se estabilizara, mostrando a continuación los resultados: en la Figura 43 se ven los ángulos, en la Figura 44 podemos ver las señales de control.

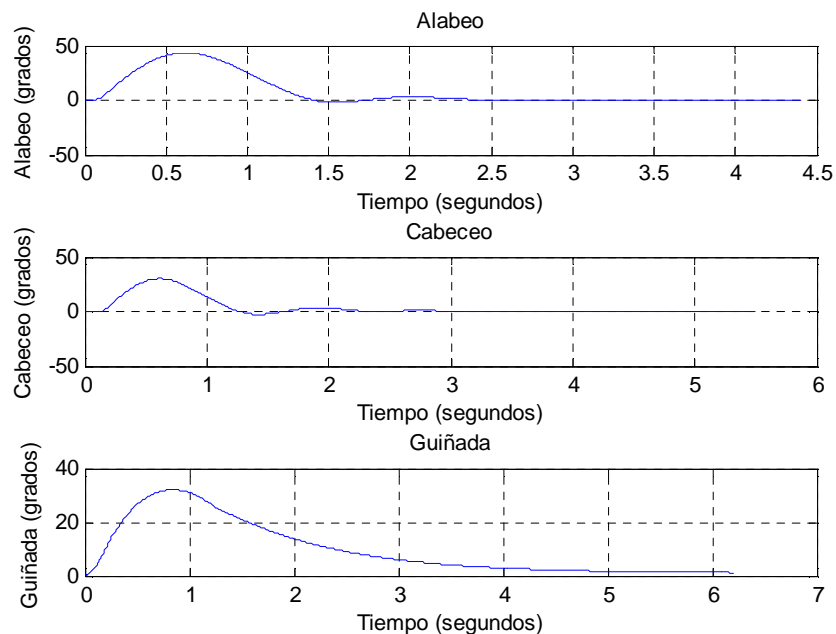


Figura 43. Estabilización de los ángulos del sistema 3D Hover de Quanser mediante un LQR

Los bloques que generan las señales de control son todas de la misma forma (apartado 3.3. Control por modos deslizantes), como la que se muestra en la ecuación (5.2)

$$u_{\theta} = -\frac{J_y}{K_f I} \left\{ \frac{(J_z - J_x)}{J_y} \psi \dot{\phi} + \lambda_1 \dot{\theta} \right\} - k_1 \text{sat}(\dot{\theta} + \lambda_1 (\theta - \theta_d)) \quad (5.2)$$

En estas señales de control cambian, únicamente, los parámetros k y λ . En el caso de los ángulos de alabeo y cabeceo, estas constantes tendrán el mismo valor, por la simetría del sistema y las llamaremos k_1 y λ_1 . Tomaran valores distintos para ángulo de guiñada y las llamaremos k_2 y λ_2 .

El subsistema que contiene la función de la señal de control se puede ver en la Figura 46. Las entradas a este subsistema son por una lado los parámetros para el control y por otro las constantes de la dinámica del sistema (momentos de inercia, distancia entre rotores) y el estado del sistema.

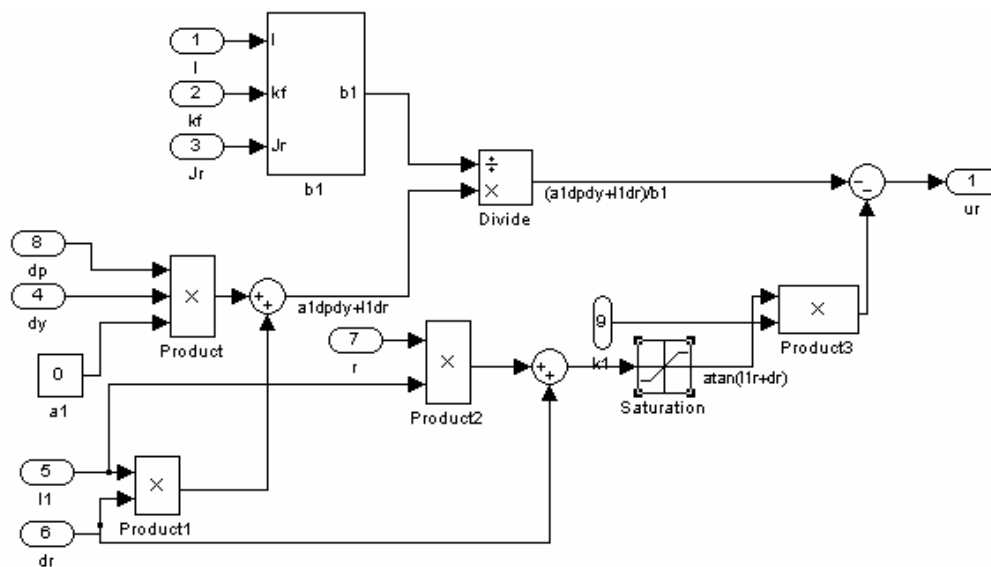


Figura 46. Subsistema de la señal de control para el cabeceo

Para obtener las siguientes gráficas se ha desestabilizado el sistema en cada uno de los ejes y se ha esperado a que el sistema se estabilizara, mostrando a continuación los resultados: en la Figura 47 se ven los ángulos, en la Figura 48 podemos ver las señales de control. Vemos que la estabilización se produce en menos de dos segundos y que oscila menos que el control LQR mostrado en la Figura 43. Esto ya lo esperábamos, pues en las simulaciones ya veíamos como el control no lineal era más rápido que el LQR.

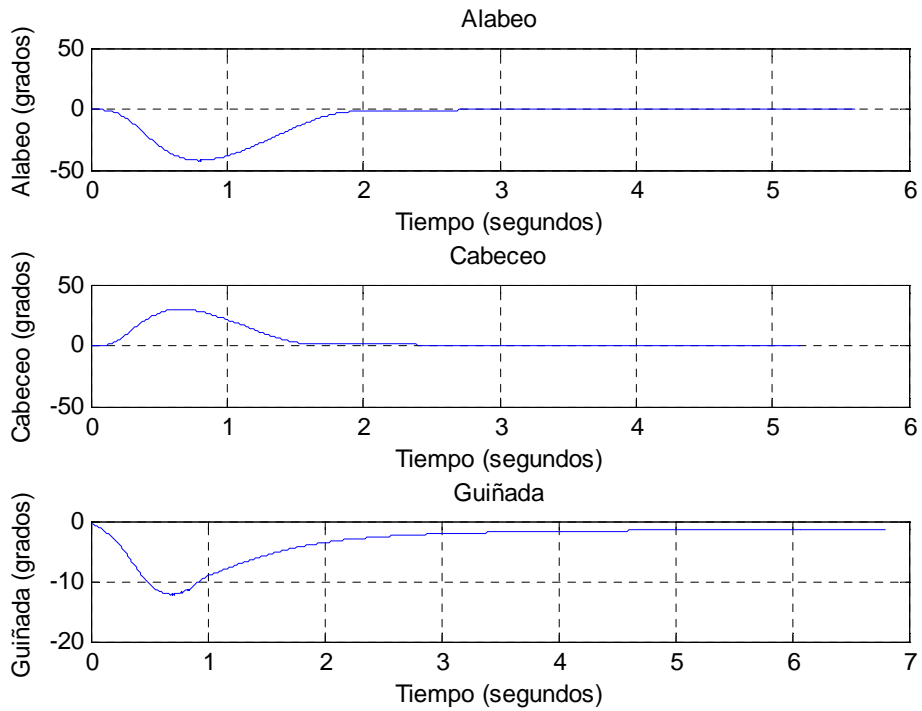


Figura 47. Estabilización de los ángulos del sistema 3D Hover de Quanser mediante un control por modos deslizantes

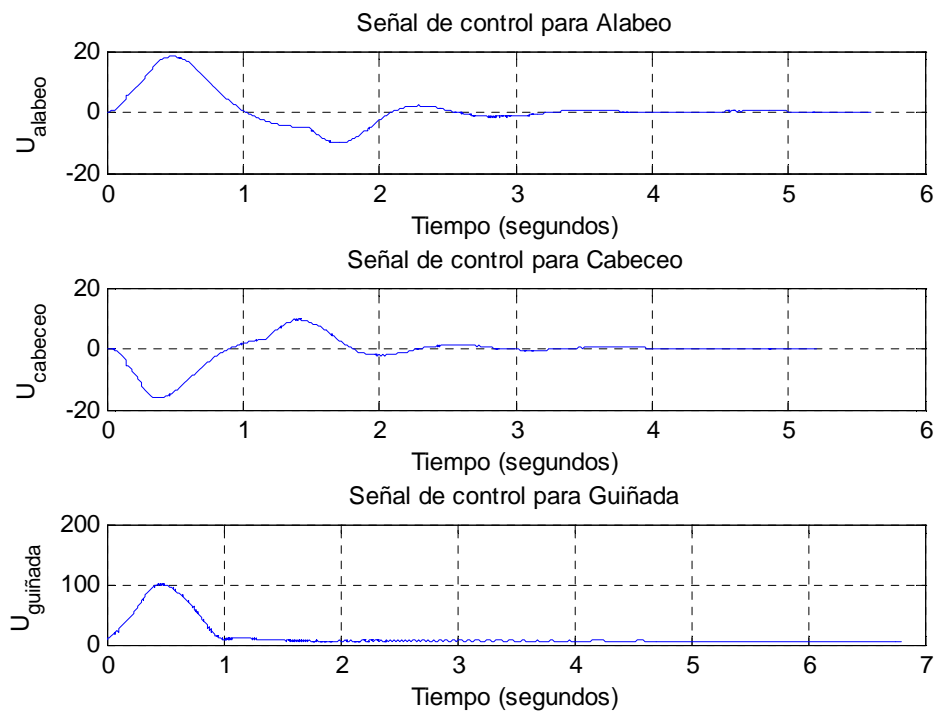


Figura 48. Señales de control para la estabilización angular del sistema 3D Hover de Quanser

5. CONCLUSIONES Y FUTUROS TRABAJOS

5.1. Conclusiones

En este trabajo de Máster se han alcanzado los objetivos propuestos. Se han analizado los motores (que lleva el prototipo experimental construido) y se ha obtenido una función de transferencia que los identifica y que se ha empleado en los modelos de simulación propuestos para el control. Además se ha logrado la estabilización del cuatrimotor mediante dos estrategias de control diferentes, una lineal y otra no lineal verificando, tanto en simulación en como las pruebas realizadas en la plataforma experimental, que ambos controladores funcionan. Para el control en posición del cuatrimotor se ha escogido finalmente como control de estabilización el desarrollado por la teoría de Modos Deslizantes, pues produce menores oscilaciones y menores tiempos de asentamiento. Este control se ha verificado en simulación y en un futuro será aplicado en un cuatrimotor real.

5.2. Trabajo futuro

El proyecto de Máster aquí presentado se enmarca dentro de la investigación que el grupo ISCAR (*Ingeniería de Sistemas de Control, Automatización y Robótica*) viene realizando trabajos sobre coordinación y colaboración de vehículos autónomos marinos y aéreos en tareas de control de difusión de sustancias en el mar (vertidos contaminantes) y búsqueda y rescate de náufragos.

El grupo dispone de varios modelos de USVs dos de los cuales pueden transportar UAVs. Los vehículos marinos tienen la posibilidad de comunicación entre ellos para colaborar en la realización de las tareas.

Para los UAVs se ha terminado la fase de construcción y la investigación se encuentra en fase de prueba de los controladores de cuatrimotores equipados, entre otros sensores, con un módulo de

Percepción Visual que ayude en la búsqueda e identificación de objetos y sustancias, así como en la aproximación y aterrizaje sobre los buques.

La coordinación y operación de los vehículos autónomos plantea un sistema de control de gran escala en el cual la tarea de control se distribuye entre varios procesadores y la comunicación entre los procesadores, sensores y actuadores es mediante canales de comunicación. Se hace necesaria la integración del control y las comunicaciones ateniéndose a las limitaciones de los canales de comunicación. Para llevar a cabo este objetivo es fundamental que los vehículos autónomos conozcan con precisión su posición y la del resto de vehículos.

En el grupo viene trabajando en temas de fusión sensorial desde hace varios años, siendo algunas de las referencias más significativas: [Besada *et al*, 2002], [Besada *et al*, 2007], [Besada *et al*, 2009a], [Besada *et al*, 2009b]. En el trabajo futuro trabajo de tesis se pretende que los últimos resultados teóricos del grupo se apliquen al problema planteado y se analicen nuevos tipos de situaciones. Para lo cual, se desarrollará una arquitectura software de control en tiempo real basada en fusión multisensorial a partir de la información de los diferentes sistemas sensoriales de los sistemas robóticos y sistemas sensoriales involucrados en el problema real (ya sea para tareas de cooperación y maniobra entre barcos, exploración o rescate de naufragos).

6. BIBLIOGRAFÍA

[Arduino] <http://www.arduino.cc/>

[Besada *et al*, 2002] E. Besada Portas, J.A. Lopez Orozco, J.M de la Cruz. "Unified fusion system based on Bayesian Networks for autonomous mobile robots". Proceedings of 5th Internacional Conference on Information Fusion, vol. 1, pp. 873, 880, July 2002.

[Besada *et al*, 2007] E. Besada-Portas, J. Lopez-Orozco, and J. de la Cruz, "Multisensor out-of-sequence data fusion for estimating the state of dynamic systems," in Proc of Information, Decision and Control. (IDC 2007), February 2007, pp. 348–353.

[Besada *et al*, 2009a] Eva Besada-Portas, Jose Antonio Lopez-Orozco, Juan Besada, Jesus M. De La Cruz (2009)"Multisensor Out Of Sequence Data Fusion for Estimating the State of Discrete Control Systems" IEEE Transactions on Automatic Control, vol. 54 (7), pp. 1728-1732.

[Besada *et al*, 2009b] Eva Besada-Portas, Jose A. Lopez-Orozco, Juan A. Besada, Jesus M. de la Cruz. "Multisensor Data Fusion for Linear Control Systems with Out-Of-Sequence and Erroneous Measurements" Automatica, en revision.

[Bouabdallah *et al*, 2004] S. Bouabdallah, P. Murrieri, R. Siegwart. *Design and Control of an Indoor Micro Quadrotor*. IEEE International Conference on Robotics and Automation, New Orleans, USA, 2004.

[Brian, 2004] Eryk Brian Nice. *Design of a four rotor hovering vehicle*. PhD Thesis. Cornell University, 2004.

[Castillo *et al*, 2004] Pedro Castillo, Alejandro Dzul and Rogelio Lozano. *Real-Time Stabilization and Tracking of a Four-Rotor Mini Rotorcraft*. IEEE transactions on control systems technology, vol.12, no.4, July 2004

[Castillo *et al*, 2007] P. Castillo, P. García, R. Lozano, P. Albertos. *Modelado y estabilización de un helicóptero con cuatro rotores*. Revista Iberoamericana de Automática e Informática Industrial. pp 41-57, ISSN: 1697-7912. Vol.4, Num 1, 2007.

- [Chen, 2003] Ming Chen. *Formation and Flight control of Affordable Quad-rotor Unmanned Air Vehicles*. PhD Thesis. The University of British Columbia, 2003.
- [Friedland,1986] B. Friedland, *Control System Design*, , Dover Publications, Inc. 1986.
- [Gordon, 2002] Professor J. Gordon Leishman. *The Bréguet-Richet Quad-Rotor Helicopter of 1907*. Ver-tiflite, 47(3):58–60, 2002
- [Hanford, 2005] Scott D. Hanford. *A small semi-autonomous rotary-wing unmanned air vehicle*. PhD thesis. The Pennsylvania State University, 2005.
- [Hespanha, 2007] João P. Hespanha *LQG/LQR controller design*, April 2007
- [Hoffmann *et al*, 2004] Gabe Hoffmann, Dev Gorur Rajnarayan, Steven L. Waslander, David Dostal, Jung Soon Jang, Claire J. Tomlin. *The stanford testbed of autonomous rotorcraft for multi agent control (STARMAC)*. Digital Avionics Systems Conference, 2004.
- [Khalil,2002] H.K. Khalil, *Nonlinear Systems*, Third Edition, Ed. Prentice-Hall, 2002.
- [Levine, 1996] William S. Levine, *The Control Handbook*, Ed. CRC Press, 1996.
- [Özgür, 2006] Arda Özgür Kivrak. Design of control systems for a quadrotor Flight vehicle equipped with inertial sensors. Master's thesis. Atılım University, December 2006.
- [Pounds *et al*, 2002a] P. Pounds, R. Mahony, P. Hynes, J. Roberts. *Design of a Four-Rotor Aerial Robot*. Australian Conference on Robotics and Automation, 2002.
- [Pounds *et al*, 2002b] P. Pounds, R. Mahony, P. Corke. *Modelling and Control of a Quad-Rotor Robot*. Australian National University, 2002
- [Quanser] www.quanser.com
- [Sánchez, 2009] David Sánchez Benítez, Marta Salas García, Jesús Manuel de la Cruz, Kinematics and Dinamics of a quadrotor, Universidad Complutense de Madrid, 2009.
- [Slotine, 1991]J-J E. Slotine, *Applied Nonlinear Control*, Weiping Li. Prentice- Hall, Inc.1991.
- [Stevens, 2003] Brian L. Stevens and Frank L. Lewis. John Wiley & Sons, *Aircraft control and Simulation*, Inc., 2003, 2nd edition.

APÉNDICE A. CINEMÁTICA Y DINÁMICA DE UN CUATRIMOTOR²

El movimiento de un cuerpo rígido posee seis grados de libertad: tres de ellos definen la posición de un punto de referencia en el cuerpo (normalmente el centro de masa, tal como será nuestro caso), y los otros tres definen la orientación del cuerpo. Puesto que cada grado de libertad toma dos variables de estado (posición y velocidad), se requerirán un total de 12 ecuaciones diferenciales para describir totalmente el movimiento del cuerpo.

El cuatrимotor es descrito por el sistema de referencia fijo al vehículo (SR^V) cuyo origen es el centro de masas (CM) del mismo y tal que su vector unitario i^V apunta hacia el frente, j^V apunta hacia la izquierda, y k^V hacia abajo. El sistema de referencia inercial (SR^I) está fijo a tierra y sus ejes apuntan hacia el norte, el este y hacia abajo, podemos verlo en la Figura 49.

La notación a seguir para las doce variables de estado que posee el cuatrимotor es la siguiente:

$\mathbf{x} = (x, y, z)$ es el vector de posición del cuatrимotor respecto a SR^I .

$\mathbf{v} = (u, v, w)$ es el vector velocidad respecto a SR^V .

$\boldsymbol{\omega} = (p, q, r)$ son las tasas de cambio de los ángulos anteriores respecto a SR^V .

$\boldsymbol{\alpha} = (\varphi, \theta, \psi)$ son los ángulos de Euler del cuatrимotor (cabeceo, alabeo y guiñada respectivamente) respecto a SR^I .

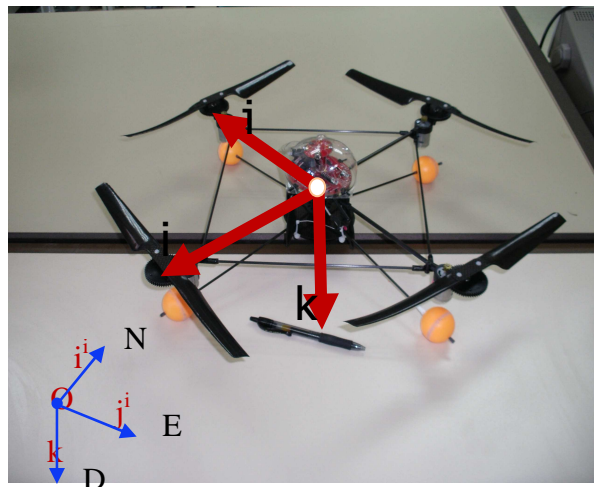


Figura 49. Sistemas de Referencia

A. Cinemática

El vector $\mathbf{x}=(x,y,z)$ está definido respecto al sistema inercial, mientras que $\mathbf{v}^V=(u,v,w)$ lo está respecto al sistema unido al cuatrимotor. Para pasar de uno a otro tendremos que usar la matriz de Euler para giros sobre los tres ejes. Suponiendo que los giros se producen (en sentido horario)

²Este apéndice ha sido extraído de [Sánchez, 2009]

primero respecto a k^v , luego respecto a j^v y finalmente respecto a i^v , la matriz de rotación tendrá la conocida forma (0.1)

$$R_{v/i}(\phi, \theta, \psi) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\phi & \sin\phi \\ 0 & -\sin\phi & \cos\phi \end{bmatrix} \begin{bmatrix} \cos\theta & 0 & -\sin\theta \\ 0 & 1 & 0 \\ \sin\theta & 0 & \cos\theta \end{bmatrix} \begin{bmatrix} \cos\psi & \sin\psi & 0 \\ -\sin\psi & \cos\psi & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (0.1)$$

Entonces la inversa tiene la forma (0.2)

$$R_{i/v} = R_{v/i}^{-1} = R_{v/i}^T = \begin{bmatrix} c(\theta)c(\psi) & s(\phi)s(\theta)c(\psi) - c(\phi)s(\psi) & c(\phi)s(\theta)c(\psi) + s(\phi)s(\psi) \\ c(\theta)s(\psi) & s(\phi)s(\theta)s(\psi) + c(\phi)c(\psi) & c(\phi)s(\theta)s(\psi) - s(\phi)c(\psi) \\ -s(\theta) & s(\phi)c(\theta) & c(\phi)c(\theta) \end{bmatrix} \quad (0.2)$$

Tendremos que, la derivada de posición en el sistema de referencia inercial tiene la forma de la ecuación (0.3)

$$\frac{d}{dt} \begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \end{bmatrix} = R_{i/v}(\phi, \theta, \psi) \begin{bmatrix} u \\ v \\ w \end{bmatrix} \quad (0.3)$$

Para relacionar (p, q, r) con $d/dt(\phi, \theta, \psi)$, consideraremos las rotaciones dadas por SR^v paso a paso, lo cual podemos ver en la ecuación (0.4)

$$\begin{bmatrix} p \\ q \\ r \end{bmatrix} = T_\phi T_\theta \begin{bmatrix} 0 \\ 0 \\ \dot{\psi} \end{bmatrix} + T_\phi \begin{bmatrix} 0 \\ \dot{\theta} \\ 0 \end{bmatrix} + \begin{bmatrix} \dot{\phi} \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} -\dot{\psi} \sin\theta \\ \dot{\psi} \sin\phi \cos\theta \\ \dot{\psi} \cos\phi \cos\theta \end{bmatrix} + \begin{bmatrix} 0 \\ \dot{\theta} \cos\phi \\ -\dot{\theta} \sin\phi \end{bmatrix} + \begin{bmatrix} \dot{\phi} \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 1 & 0 & -\sin\theta \\ 0 & \cos\phi & \sin\phi \cos\theta \\ 0 & -\sin\phi & \cos\phi \cos\theta \end{bmatrix} \begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} \quad (0.4)$$

E invirtiendo la ecuación (0.4), se obtiene la relación (0.5)

$$\begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} = \begin{bmatrix} 1 & s\phi \tan\theta & c\phi \tan\theta \\ 0 & c\phi & -s\phi \\ 0 & s\phi/c\theta & c\phi/c\theta \end{bmatrix} \begin{bmatrix} q \\ r \end{bmatrix} \quad (0.5)$$

B. Dinámica

De la segunda ley de Newton: $M_T \frac{dv^i}{dt_i} = f$. Al ser SR^v un sistema que rota y se desplaza respecto a SR^i , tomaremos la ecuación de Coriolis (0.6)

$$M_T \frac{dv^i}{dt_i} = M_T \left(\frac{dv}{dt_i} + \omega \times v \right) = f \quad (0.6)$$

donde M_T es la masa total del cuatrimotor ($M_T = M + 4m$, ver Figura 51) y f representa las fuerzas a las que se ve sometido el cuatrimotor.

Tenemos que $\mathbf{v} = (u, v, w)$, así es de la forma que muestra (0.7)

$$\begin{bmatrix} \dot{u} \\ \dot{v} \\ \dot{w} \end{bmatrix} + \begin{bmatrix} i & j & k \\ P & Q & R \\ U & V & W \end{bmatrix} = \frac{f}{M_T} \rightarrow \begin{bmatrix} \dot{u} \\ \dot{v} \\ \dot{w} \end{bmatrix} = - \begin{bmatrix} i & j & k \\ P & Q & R \\ U & V & W \end{bmatrix} + \frac{f}{M_T} \rightarrow \begin{bmatrix} \dot{u} \\ \dot{v} \\ \dot{w} \end{bmatrix} = \begin{bmatrix} RV-QW \\ PW-RU \\ QU-PV \end{bmatrix} + \frac{f}{M_T} \quad (0.7)$$

Si analizamos las fuerzas a las que estará sometido el cuatrimotor, éstas serán la sustentación producida por cada motor, la ejercida por la gravedad, y los correspondientes rozamientos. Analicemos cada una a continuación.

A.1. Efecto de los motores

La fuerza total ascendente que sufra el cuatrimotor será la suma de las cuatro fuerzas producidas por los distintos motores. Desde el punto de vista del cuatrimotor, la sustentación tendrá la forma: $\mathbf{F}^v = -(0, 0, F_1+F_2+F_3+F_4) = -(0, 0, \text{Throttle})$. El signo negativo es consecuencia de haber tomado el sentido positivo del eje k^v hacia abajo. El giro de cada rotor se puede ver en la Figura 50

Cada una de estas fuerzas es proporcional al pulso eléctrico que se le envía a cada motor (Ω_i). Esto se puede expresar como (0.8)

$$F_i = k\Omega_i \quad i = 1, 2, 3, 4 \quad (0.8)$$

La constante k se calcula experimentalmente.

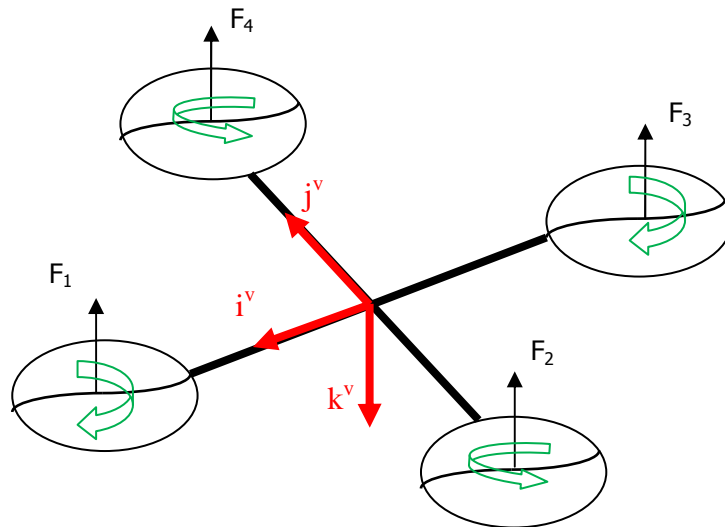


Figura 50. Fuerzas ejercidas por los rotores

A.2. Gravedad

Por su parte, desde SR^i , la fuerza ejercida por la gravedad tiene la forma (0.9)

$$\mathbf{F}_g^i = (0, 0, M_T g) \quad (0.9)$$

Desde el sistema adjunto al vehículo la fuerza de la gravedad toma la forma de la ecuación (0.10)

$$F_g^v = R_{vi} F_g^i = \begin{bmatrix} c(\theta)c(\psi) & c(\theta)s(\psi) & -s(\theta) \\ s(\phi)s(\theta)c(\psi) - c(\phi)s(\psi) & s(\phi)s(\theta)s(\psi) + c(\phi)c(\psi) & s(\phi)c(\theta) \\ c(\phi)s(\theta)c(\psi) + s(\phi)s(\psi) & c(\phi)s(\theta)s(\psi) - s(\phi)c(\psi) & c(\phi)c(\theta) \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ M_T g \end{bmatrix}$$

$$F_g^v = \begin{bmatrix} -s(\theta)M_T g \\ s(\phi)c(\theta)M_T g \\ c(\phi)c(\theta)M_T g \end{bmatrix} \quad (0.10)$$

A.3. Rozamientos

La fuerza de rozamiento que sufre el vehículo, en cada eje de coordenadas, tiene la misma expresión [3.11]

$$F_{ri} = \frac{1}{2} C_i A_c \rho |i| i \quad i = u, v, w, p, q \quad (0.11)$$

El coeficiente C_i depende de varios factores ([Stevens, 2003]), pero para nuestro caso nos bastará considerarlo constante.

Como se puede comprobar, los términos correspondientes a estos rozamientos son muy pequeños, por lo que, aunque se han incorporado en la dinámica del cuatrimotor, podrían ser despreciados sin grandes efectos en el comportamiento del modelo simulado en estado de equilibrio.

Con esto, nos quedará la ecuación (0.12)

$$\begin{bmatrix} \dot{u} \\ \dot{v} \\ \dot{w} \end{bmatrix} = \begin{bmatrix} RV-QW \\ PW-RU \\ QU-PV \end{bmatrix} + \begin{bmatrix} -s(\theta)g - \frac{F_{rx}}{M_T} \\ s(\phi)c(\theta)g - \frac{F_{ry}}{M_T} \\ c(\phi)c(\theta)g - \frac{(\text{Throttle}-F_{rz})}{M_T} \end{bmatrix} \quad (0.12)$$

A.4. Rotaciones

Para el movimiento rotacional del cuatrimotor tendremos que: $\frac{d\mathbf{M}}{dt} = \mathbf{T}$ Donde \mathbf{M} es el momento angular y $\mathbf{T} = (\tau_\phi, \tau_\theta, \tau_\psi)$ el momento aplicado.

Por una parte, $\mathbf{M} = \mathbf{J} \cdot \boldsymbol{\omega}$, con $\mathbf{J} = \begin{bmatrix} J_x & -J_{xy} & -J_{xz} \\ -J_{yx} & J_y & -J_{yz} \\ -J_{zx} & -J_{zy} & J_z \end{bmatrix}$ la matriz de inercia, que si consideramos al

cuatrimotor como un elemento simétrico respecto de sus ejes, tomará la forma diagonal mostrada en (0.13)

$$\mathbf{J} = \begin{bmatrix} J_x & 0 & 0 \\ 0 & J_y & 0 \\ 0 & 0 & J_z \end{bmatrix} \quad (0.13)$$

Si consideramos al cuatrimotor como una esfera central de masa M y radio R de la que se proyectan 4 varillas sin masa, de longitud l , en cuyos extremos se encuentra una masa puntual m , que representa al conjunto del motor y hélice (Figura 51), tendremos que los momentos de inercia se expresan en la ecuación (0.14)

$$\begin{aligned} J_i &= \frac{2}{5}MR^2 + 2l^2m, \text{ para } i=x, y \\ J_z &= \frac{2}{5}MR^2 + 4l^2m \end{aligned} \quad (0.14)$$

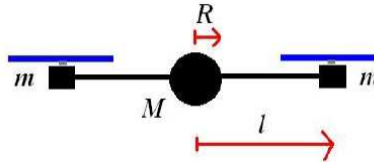


Figura 51. Disposición del cuatrimotor

Por otra parte, los momentos aplicados ($\tau_\phi, \tau_\theta, \tau_\psi$) tomarán distintas formas, al depender cada uno de una configuración distinta de los motores.

La rotación en torno a ϕ (alabeo) se deberá a una diferencia entre las fuerzas F_2 y F_4 . Concretamente tendremos la ecuación (0.15)

$$\tau_\phi = l(F_2 - F_4) \quad (0.15)$$

donde l es la distancia del centro del cuatrimotor a cualquiera de los rotores.

La rotación en torno a θ (cabeceo) se deberá a una diferencia entre las fuerzas F_1 y F_3 .

Concretamente tendremos la ecuación (0.16)

$$\tau_\theta = l(F_1 - F_3) \quad (0.16)$$

La rotación del cuatrimotor entorno a su propio eje (guiñada) se deberá a una diferencia en el momento aplicado por los rotores delantera y trasero, y los laterales, expresado por la ecuación (0.17)

$$\tau_\psi = \tau_1 + \tau_3 - \tau_2 - \tau_4 = k'(\Omega_1 + \Omega_3 - \Omega_2 - \Omega_4) \quad (0.17)$$

donde la constante k' se calcula experimentalmente.

Al ser SR^v un sistema que rota y se desplaza respecto a SR^i , tomaremos la ecuación de

Coriolis:
$$\frac{dM}{dt_i} = \frac{dM}{dt_v} + \omega \times M = T$$

De esta manera llegamos a las *ecuaciones de Euler* que describen como las componentes del vector velocidad angular evolucionan en el tiempo, en respuesta a los momentos aplicados [Friedland,1986], ecuación (0.18)

$$\frac{d(J \cdot \omega)}{dt_v} + \omega \times (J \cdot \omega) = (\tau_\phi, \tau_\theta, \tau_\psi) \Rightarrow J \begin{bmatrix} \dot{p} \\ \dot{q} \\ \dot{r} \end{bmatrix} + \begin{bmatrix} i & j & k \\ p & q & r \\ J_x p & J_y q & J_z r \end{bmatrix} = \begin{bmatrix} \tau_\phi \\ \tau_\theta \\ \tau_\psi \end{bmatrix}$$

$$\begin{bmatrix} \dot{p} \\ \dot{q} \\ \dot{r} \end{bmatrix} = J^{-1} \left\{ \begin{bmatrix} \tau_\phi \\ \tau_\theta \\ \tau_\psi \end{bmatrix} - \begin{bmatrix} i & j & k \\ p & q & r \\ J_x p & J_y q & J_z r \end{bmatrix} \right\} = \begin{bmatrix} \tau_\phi/J_x \\ \tau_\theta/J_y \\ \tau_\psi/J_z \end{bmatrix} + \begin{bmatrix} \frac{(J_y - J_z)}{J_x} qr \\ \frac{(J_z - J_x)}{J_y} pr \\ \frac{(J_x - J_y)}{J_z} pq \end{bmatrix} \quad (0.18)$$

Si introducimos el rozamiento que sufrirá el vehículo en su rotación en la ecuación (0.18) obtendremos la ecuación final (0.19)

$$\begin{bmatrix} \dot{p} \\ \dot{q} \\ \dot{r} \end{bmatrix} = \begin{bmatrix} (\tau_\phi - F_{r\phi})/J_x \\ (\tau_\theta - F_{r\theta})/J_y \\ \tau_\psi/J_z \end{bmatrix} + \begin{bmatrix} \frac{(J_y - J_z)}{J_x} qr \\ \frac{(J_z - J_x)}{J_y} pr \\ \frac{(J_x - J_y)}{J_z} pq \end{bmatrix} \quad (0.19)$$

Las ecuaciones (0.3), (0.5), (0.12) y (0.19) forman el sistema de ecuaciones que describen al cuatrimotor. Las volvemos a escribir agrupadas:

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \end{bmatrix} = \begin{bmatrix} c(\theta)c(\psi) & s(\phi)s(\theta)c(\psi) - c(\phi)s(\psi) & c(\phi)s(\theta)c(\psi) + s(\phi)s(\psi) \\ c(\theta)s(\psi) & s(\phi)s(\theta)s(\psi) + c(\phi)c(\psi) & c(\phi)s(\theta)s(\psi) - s(\phi)c(\psi) \\ -s(\theta) & s(\phi)c(\theta) & c(\phi)c(\theta) \end{bmatrix} \begin{bmatrix} u \\ v \\ w \end{bmatrix}$$

$$\begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} = \begin{bmatrix} 1 & s\phi \tan\theta & c\phi \tan\theta \\ 0 & c\phi & -s\phi \\ 0 & s\phi/c\theta & c\phi/c\theta \end{bmatrix} \begin{bmatrix} q \\ r \\ r \end{bmatrix}$$

$$\begin{bmatrix} \dot{u} \\ \dot{v} \\ \dot{w} \end{bmatrix} = \begin{bmatrix} RV-QW \\ PW-RU \\ QU-PV \end{bmatrix} + \begin{bmatrix} -s(\theta)g - \frac{F_{rx}}{M_T} \\ s(\phi)c(\theta)g - \frac{F_{ry}}{M_T} \\ c(\phi)c(\theta)g - \frac{(\text{Throttle}-F_{rz})}{M_T} \end{bmatrix}$$

$$\begin{bmatrix} \dot{p} \\ \dot{q} \\ \dot{r} \end{bmatrix} = \begin{bmatrix} (\tau_\phi - F_{r\phi})/J_x \\ (\tau_\theta - F_{r\theta})/J_y \\ \tau_\psi/J_z \end{bmatrix} + \begin{bmatrix} \frac{(J_y - J_z)}{J_x} qr \\ \frac{(J_z - J_x)}{J_y} pr \\ \frac{(J_x - J_y)}{J_z} pq \end{bmatrix} \quad \text{con} \quad \begin{aligned} \tau_\phi &= l(F_2 - F_4) = lk(\Omega_2 - \Omega_4) \\ \tau_\theta &= l(F_1 - F_3) = lk(\Omega_1 - \Omega_3) \\ \tau_\psi &= \tau_1 + \tau_3 - \tau_2 - \tau_4 = k'(\Omega_1 + \Omega_3 - \Omega_2 - \Omega_4) \\ J_i &= \frac{2}{5}MR^2 + 2I^2m, \text{ para } i=x, y \\ J_z &= \frac{2}{5}MR^2 + 4I^2m \end{aligned}$$