



UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA



ESCUELA TÉCNICA
SUPERIOR INGENIEROS
INDUSTRIALES VALENCIA

TRABAJO FIN DE GRADO EN INGENIERÍA EN TECNOLOGÍAS INDUSTRIALES

**DISEÑO, IMPLEMENTACIÓN Y EVALUACIÓN DE
ALGORITMOS DE CONTROL, PARA MEJORAR LA
ESTABILIDAD DE UN SISTEMA QUADROTOR CON
INCERTIDUMBRE DE MODELADO Y
PERTURBACIONES EXTERNAS**

AUTOR: JOAQUÍN ROYO MIQUEL

TUTOR: PEDRO GARCÍA GIL

Curso Académico: 2017/2018

Agradecimientos

A Vicente, Joan y el resto de mis
compañeros de laboratorio, por estar
ahí todos los días.

A mi tutor, Pedro García Gil, por las
clases de TAU y por darme la
oportunidad de hacer este trabajo.

A mis compañeros de la carrera,
ahora buenos amigos, que
consiguieron que tuviese ganas de
pisar la universidad.

A mi familia, que me apoya, y sobre
todo que me aguanta... y que me paga
los estudios.

A todos esos amigos que me habéis
pedido venir a verme defender el TFG
y os he dado largas.

Al G.S. Altaïr.

Gracias.

Resumen

El presente documento conforma la memoria del trabajo de fin de grado “Diseño, implementación y evaluación de algoritmos de control, para mejorar la estabilidad de un sistema quadrotor con incertidumbre de modelado y perturbaciones externas” de la titulación de Grado en Ingeniería de las Tecnologías Industriales cursada en la Escuela Técnica Superior de Ingenieros Industriales de la Universidad Politécnica de Valencia durante los años 2014-2018.

El objetivo del proyecto es el estudio, diseño y aplicación de diferentes estrategias de control, utilizadas en la industria, a la estabilización de un vehículo quadrotor y su análisis comparativo.

Para ello se dividirá el trabajo en dos partes: diseño y simulación de los diferentes sistemas de control, y validación experimental sobre el prototipo de un quadrotor.

Como base para el análisis, partiendo de las ecuaciones generales que definen la dinámica de un quadrotor, se obtendrá un modelo simplificado que aproxime el comportamiento real del sistema en torno a un punto de funcionamiento y se validará a partir de una identificación experimental de nuestro sistema.

Una vez obtenido un modelo satisfactorio del funcionamiento del sistema, se diseñarán los algoritmos de control aplicando diversas técnicas conocidas, los cuales se ajustarán y validarán a través de simulaciones considerando el modelo obtenido anteriormente. Posteriormente, se implementarán en una plataforma física disponible consistente en un sistema quadrotor anclado a una rótula, se realizará un último ajuste, y se comparará el funcionamiento y la robustez de todas ellas identificando aquella que destaque.

La técnica elegida será implementada como algoritmo de control para el quadrotor en vuelo libre y sobre la misma podrá aplicarse un control de posición en cascada que dé funcionalidad al vehículo.

Palabras Clave: PID, observadores, ESO, UDE, quadrotor, control.

Índice general

| | |
|---|----|
| Agradecimientos | 2 |
| Resumen..... | 4 |
| Índice general..... | 6 |
| Índice de figuras..... | 8 |
| Índice de tablas..... | 9 |
| Índice de ecuaciones..... | 10 |
| | |
| MEMORIA..... | 12 |
| 1. Introducción | 14 |
| 1.1. Objetivos del proyecto..... | 14 |
| 1.2. Estructura del documento | 15 |
| 2. Revisión bibliográfica..... | 16 |
| 2.1. Introducción | 16 |
| 2.2. Antecedentes históricos del quadrotor | 16 |
| 2.3. Técnicas de control para sistemas inestables..... | 18 |
| 2.4. Aplicaciones actuales de los drones..... | 19 |
| 2.5. Normativa | 20 |
| 2.6. Motivación y justificación del proyecto..... | 21 |
| 3. Descripción de la Plataforma..... | 22 |
| 3.1. El quadrotor..... | 22 |
| 3.2. La fuente de alimentación..... | 26 |
| 3.3. El ordenador de tierra | 27 |
| 3.4. El banco de ensayos..... | 27 |
| 3.5. El sistema <i>Optitrack</i> | 28 |
| 4. Modelado del sistema..... | 30 |
| 4.1. Funcionamiento del quadrotor..... | 30 |
| 4.2. Modelo Teórico del quadrotor | 31 |
| 4.3. Modelo lineal aproximado:..... | 36 |
| 4.4. Quadrotor en <i>cruz</i> y en <i>equis</i> | 37 |
| 5. Identificación | 40 |
| 5.1. Método empleado:..... | 40 |
| 5.2. Identificación de la orientación en <i>pitch</i> y <i>roll</i> | 41 |
| 5.3. Identificación de la orientación en <i>yaw</i> | 43 |

| | | |
|--|--|----|
| 6. | Sistema de control de la orientación | 44 |
| 6.1. | Introducción | 44 |
| 6.2. | Criterios de diseño | 44 |
| 6.3. | Métodos clásicos | 46 |
| 6.4. | Estimación del estado | 53 |
| 6.5. | Realimentación del estado | 58 |
| 6.6. | Conclusiones del apartado:..... | 62 |
| 7. | Sistema de control de la posición | 64 |
| 7.1. | Contextualización | 64 |
| 7.2. | Algoritmo empleado..... | 65 |
| 8. | Conclusiones..... | 68 |
| Anexo I – Simulaciones programadas | | 70 |
| | Subsistema de rotación..... | 70 |
| | Subsistema de traslación | 72 |
| PRESUPUESTO | | 74 |
| 1. | Introducción | 76 |
| 2. | Cuadro de precios de mano de obra | 76 |
| 3. | Cuadro de precios de maquinaria..... | 76 |
| 4. | Cuadro de precios unitarios..... | 77 |
| 5. | Cuadro de precios descompuestos..... | 78 |
| 6. | Presupuesto Base de Licitación | 80 |
| Bibliografía..... | | 82 |

Índice de figuras

| | |
|---|----|
| Figura 1 – Representación artística del Taketombo [35] | 16 |
| Figura 2 – Boceto del Tornillo Aéreo de Leonardo da Vinci [34] | 17 |
| Figura 3 – Helicóptero de Bothezat [3] | 17 |
| Figura 4 – Autogiro Cierva C-40 [4] | 17 |
| Figura 5 – Modelo de quadrotor Amazon Prime Air [11]. | 20 |
| Figura 6 – El Quadrotor | 22 |
| Figura 7 – Raspberry Pi 2 [14] | 23 |
| Figura 8 – Esquema del filtro complementario | 24 |
| Figura 9 – La IMU | 25 |
| Figura 10 – Motor del dron (detalle) | 25 |
| Figura 11 – Proyecto QuadControl en el programa Code::Blocks | 26 |
| Figura 12 – Fuente de alimentación | 27 |
| Figura 13 – Banco de ensayos o rótula | 28 |
| Figura 14 – Reconstrucción de la posición del dron (dcha.) en el programa Motive (izqda.) | 28 |
| Figura 15 – Cámara Flex:V100 del equipo Optitrack del laboratorio (detalle) | 29 |
| Figura 16 – Esquema ilustrativo del quadrotor en cruz. | 30 |
| Figura 17 – Estructuras de control de un cuadricóptero | 37 |
| Figura 18 – Bucle de control para la identificación del sistema quadrotor | 40 |
| Figura 19 - Gráficas comparativas del modelo con los resultados de los ensayos realizados | 42 |
| Figura 20 - Gráfica comparativas del modelo con el resultado del ensayo sobre el yaw | 43 |
| Figura 21 – Esquema general de un sistema de control en bucle cerrado. | 44 |
| Figura 22 – Quadrotor anclado a la rótula del banco de pruebas. | 45 |
| Figura 23 – Estructura del control PID | 47 |
| Figura 24 – Gráfica del ensayo del control PD | 48 |
| Figura 25 – Gráfica del ensayo del control PID | 49 |
| Figura 26 - Estructura del control PID con filtrado de la referencia | 50 |
| Figura 27 – Gráfica del ensayo del control PID con filtrado en la referencia | 50 |
| Figura 28 - Estructura del control PID de dos grados de libertad | 51 |
| Figura 29 – Gráfica del ensayo del control PID con dos grados de libertad | 52 |
| Figura 30 - Estructura de un observador | 53 |
| Figura 31 – Esquema de la prealimentación de la perturbación | 54 |
| Figura 32 - Estructura del control PD + ESO | 55 |
| Figura 33 - Simulación del potencial del ESO (sin ruido en la medida) | 56 |
| Figura 34 – Efecto del ruido blanco sobre el control ESO y UDE | 56 |
| Figura 35 – Gráfica del ensayo del control PD + ESO | 57 |
| Figura 36 – Gráfica del ensayo del control PID de 2 g.d.l. con medición de la derivada | 58 |
| Figura 37 - Estructura del control PD + UDE | 59 |
| Figura 38 – Gráfica del ensayo del control PD + UDE | 60 |
| Figura 39 – Gráfica del ensayo del control PD+UDE del yaw | 63 |
| Figura 40 - Estructura del quadrotor en equis con el yaw fijo | 64 |
| Figura 41 – Esquema del control de posición en los ejes X e Y (PD+UDE) | 65 |
| Figura 42 – Gráfica del ensayo en vuelo para el eje X | 66 |
| Figura 43 – Gráfica del ensayo en vuelo para el eje Y | 66 |
| Figura 44 – Gráfica del control del yaw | 67 |
| Figura 45 – Gráfica del ensayo en vuelo para el eje Z | 67 |
| Figura 46 – Dron en vuelo hover estabilizado sobre el origen de coordenadas | 68 |
| Figura 47 – Simulación del control PID con filtro en la referencia (continuo) | 70 |
| Figura 48 – Simulación del control PID con dos grados de libertad (continuo) | 71 |
| Figura 49 – Simulación del control PD+ESO (continuo) | 71 |
| Figura 50 – Simulación del control PD+UDE (continuo) | 71 |
| Figura 51 – Simulación del control implementado en código (discreto) | 72 |
| Figura 52 – Simulación del control de los dos subsistemas en cascada (discreto) | 73 |

Índice de tablas

MEMORIA

| | |
|--|-----------|
| <i>Tabla 1 – Parámetros del controlador PD</i> | <i>47</i> |
| <i>Tabla 2 – Parámetros del controlador PID</i> | <i>48</i> |
| <i>Tabla 3 – Controlador PID de dos grados de libertad</i> | <i>51</i> |
| <i>Tabla 4 – Parámetros del controlador PD + ESO</i> | <i>57</i> |
| <i>Tabla 5 – Parámetros del controlador PD + UDE</i> | <i>60</i> |
| <i>Tabla 6 – Parámetros del controlador PD+UDE del yaw</i> | <i>62</i> |
| <i>Tabla 7 – Parámetros del control PD+UDE en los ejes X e Y</i> | <i>65</i> |

PRESUPUESTO

| | |
|---|-----------|
| <i>Tabla 8 – Cuadro de precios de mano de obra</i> | <i>76</i> |
| <i>Tabla 9 – Cuadro de precios de la maquinaria</i> | <i>77</i> |
| <i>Tabla 10 – Cuadro de precios unitarios</i> | <i>77</i> |
| <i>Tabla 11 – Cuadro de precios descompuestos</i> | <i>78</i> |
| <i>Tabla 12 – Presupuesto Base de Licitación</i> | <i>80</i> |

Índice de ecuaciones

| | |
|---|-----------|
| <i>Ecuación 1 – Matriz de inercia diagonal del quadrotor.....</i> | <i>31</i> |
| <i>Ecuación 2 – Coordenadas generalizadas de orientación.....</i> | <i>31</i> |
| <i>Ecuación 3 – Coordenadas generalizadas de la posición.....</i> | <i>31</i> |
| <i>Ecuación 4 – Matriz de rotación de un sólido.....</i> | <i>32</i> |
| <i>Ecuación 5 – Rotación desde el sistema cuerpo hasta el sistema inercial.....</i> | <i>32</i> |
| <i>Ecuación 6 – Relación entre las derivadas de los ángulos de Tait-Bryan y las velocidades angulares medidas por la IMU.....</i> | <i>32</i> |
| <i>Ecuación 7 – Fórmula de Euler-Lagrange.....</i> | <i>32</i> |
| <i>Ecuación 8 – Fuerzas y pares generalizados.....</i> | <i>33</i> |
| <i>Ecuación 9 – Matriz de coordenadas generalizadas.....</i> | <i>33</i> |
| <i>Ecuación 10 – Función lagrangiana.....</i> | <i>34</i> |
| <i>Ecuación 11 – Ecuación del modelo traslacional.....</i> | <i>34</i> |
| <i>Ecuación 12 – Ecuación del modelo traslacional desarrollada.....</i> | <i>34</i> |
| <i>Ecuación 13 – Ecuación del modelo rotacional.....</i> | <i>35</i> |
| <i>Ecuación 14 – Modelo simplificado.....</i> | <i>36</i> |
| <i>Ecuación 15 – Modelo simplificado en notación de Lagrange.....</i> | <i>37</i> |
| <i>Ecuación 16 – Modelo simplificado en Espacio de Estados.....</i> | <i>37</i> |
| <i>Ecuación 17 – Función de transferencia del modelo en bucle cerrado para la identificación.....</i> | <i>41</i> |
| <i>Ecuación 18 – F. de transferencia obtenida experimentalmente en la identificación de roll y pitch.....</i> | <i>41</i> |
| <i>Ecuación 19 – F. de transferencia obtenida experimentalmente en la identificación del yaw.....</i> | <i>43</i> |
| <i>Ecuación 20 – Acción proporcional (PID).....</i> | <i>46</i> |
| <i>Ecuación 21 – Acción integral (PID).....</i> | <i>46</i> |
| <i>Ecuación 22 – Acción derivada (PID).....</i> | <i>46</i> |
| <i>Ecuación 23 – Función de transferencia del controlador PID.....</i> | <i>47</i> |
| <i>Ecuación 24 – Función de transferencia del controlador PID de 2gdl.....</i> | <i>51</i> |
| <i>Ecuación 25 – Forma del estado estimado por el Observador de Luemberger.....</i> | <i>53</i> |
| <i>Ecuación 26 – Error teórico cometido por el Observador de Luemberger.....</i> | <i>54</i> |
| <i>Ecuación 27 – Ecuación del estado del sistema incluyendo la perturbación.....</i> | <i>54</i> |
| <i>Ecuación 28 – Forma del estado estimado por el E.S.O.....</i> | <i>55</i> |
| <i>Ecuación 29 – Ecuación de estimación de incertidumbre del U.D.E.....</i> | <i>59</i> |
| <i>Ecuación 30 – Ecuación del U.D.E. particularizada al sistema quadrotor.....</i> | <i>59</i> |

Parte 1:

MEMORIA

Introducción

1.1. Objetivos del proyecto

Los reguladores tipo PID son, sin lugar a duda, los de mayor implantación dentro de un entorno industrial.

Sin embargo, la cada vez mayor automatización de procesos más complejos hace necesaria la aparición de nuevos algoritmos de control que, conservando las principales ventajas de los reguladores tipo PID, se adapten mejor a sistemas de esta índole, como pueden ser aquellos con una dinámica más rápida o incluso inestables.

Este es el caso de los vehículos aéreos no tripulados, y en concreto de los quadrotors. La mayoría de los autopilotos comerciales, todavía usan reguladores tipo PID (con diferentes configuraciones), sin embargo, el requisito de dotar a estos dispositivos, no solo de la capacidad de vuelo *hover*, sino también de poder ejecutar maniobras agresivas, hace necesario el desarrollo de nuevos controladores con mayor robustez y mejores prestaciones.

En concreto, en este TFG, se busca evaluar las distintas configuraciones de reguladores PIDs empleadas en la industria moderna, y compararlas con algoritmos basados en el observador de Luenberger y en un estimador de incertidumbre de modelado y perturbaciones externas (UDE).

Para ello será necesario realizar las siguientes tareas:

1. A partir de un Quadrotor diseñado previamente en el *Instituto Universitario de Automática e Informática Industrial (Ai2)*, entender el funcionamiento de los elementos HW y SW que lo componen.
2. Realizar el modelado e identificación del prototipo considerado
3. En base al modelo considerado, realizar el diseño y simulación de los diferentes algoritmos de control objetos del estudio de este trabajo
4. Discretización de los algoritmos considerados, para su implementación computacional en el sistema SW considerado (autopiloto desarrollado en el Ai2).
5. Ajuste experimental y validación de los diferentes algoritmos estudiados
6. Comparación de los resultados, tanto a nivel de simulación como experimental, y conclusiones sobre qué algoritmo, definidos previamente unos índices óptimos, consigue mejores prestaciones para el prototipo considerado.

La realización de este proyecto, se plantea para como vía para poner en práctica y profundizar en los conocimientos adquiridos en varias disciplinas del Grado en Ingeniería en Tecnologías Industriales, tales como los aprendidos en las asignaturas de Sistemas Automáticos y Tecnología Automatización, así como su integración y coincidencia con otras disciplinas, como conocimientos de matemáticas, electrónica y programación, necesarias para poder realizar la implementación computacional de los reguladores diseñados.

1.2. Estructura del documento

La memoria del trabajo comprende los siguientes capítulos:

- Capítulo 1 – Introducción
Expone un breve preámbulo de la memoria del trabajo de fin de grado y su organización, describiendo los capítulos que la forman.
- Capítulo 2 – Estado del Arte
Contextualiza el trabajo realizado en la situación actual de los vehículos aéreos no tripulados y los avances realizados en el diseño de este tipo de aeronaves. Se explica como dicha situación motiva la realización del presente trabajo.
- Capítulo 3 – Descripción de la plataforma
Describe el sistema quadrotor empleado en el trabajo (hardware y software), el banco de trabajo que se ha utilizado para realizar los ensayos y el sistema de medida *Optitrack*.
- Capítulo 4 – Modelado del Sistema
En este capítulo se explica el cálculo de la dinámica del quadrotor en vuelo. A partir de las ecuaciones obtenidas se obtiene un modelo simplificado de su comportamiento que se utiliza como base para los diseños realizados.
- Capítulo 5 – Identificación
Particulariza el modelo obtenido en el apartado anterior al dron con el que trabajaremos a partir de los resultados de ensayos aplicados sobre el mismo. En este apartado se hacen patentes las limitaciones del modelo teórico.
- Capítulo 6 – Sistemas de control para la orientación
Apoyados en el modelo se aplican distintos algoritmos de control y se analizan los resultados que estos ofrecen en simulación y en ensayos sobre el sistema quadrotor anclado en una rótula.
- Capítulo 7 – Sistemas de control para la posición
Se valida el algoritmo elegido en el capítulo anterior sobre el dron en vuelo libre a la vez que se implementa una segunda capa de control para el gobierno de la ubicación del vehículo.
- Capítulo 8 – Conclusiones
Análisis de los resultados obtenidos y deducciones realizadas. Propuestas para proyectos futuros.

Revisión bibliográfica

2.1. Introducción

Un quadrotor o cuadricóptero es un vehículo volador que es capaz de elevarse y desplazarse debido a la acción de cuatro propulsores anclados en los extremos de una estructura cruciforme. De ellos, dos giran en sentido horario y los otros dos en sentido antihorario.

La aeronave es capaz de elevarse en el aire gracias a la suma de los empujes proporcionados por dichos rotores y de maniobrar mediante los momentos generados por la diferencia de empujes entre motores complementarios. Esto constituye una de las principales ventajas de los cuadricópteros, pues, a diferencia de los helicópteros que requieren de un plato móvil para variar el ángulo de las aspas y controlar el movimiento del vehículo, estos poseen una mayor simplicidad mecánica con hélices de paso fijo.

Sin embargo, la configuración mencionada, pese a otorgar al vehículo una alta maniobrabilidad, también dificulta en gran medida su control debido al acoplamiento de la dinámica de los cuatro propulsores.

Esto se explicará con más detalle en el [Capítulo 4 - Modelo del Sistema](#).

Los quadrotores se clasifican como UAVs (*Unmanned Aerial Vehicle*), aeronaves capaces de llevar a cabo misiones sin tripulación a bordo; cuyo vuelo puede ser dirigido por control remoto o a través de controladores autónomos (AAV) sin necesidad de intervención humana.

2.2. Antecedentes históricos del quadrotor

2.2.1. Breve historia de la aviación de ala giratoria

Fuentes: [1], [2].

La idea de una aeronave de despegue vertical se remonta a los Taketombos (竹とんぼ - “*Libélula de Bambú*”), juguete oriental originado en China en torno al 400 a.C. Su diseño fue probablemente inspirado por la observación de semillas de árboles como el sicomoro.



Figura 1 – Representación artística del Taketombo [35]

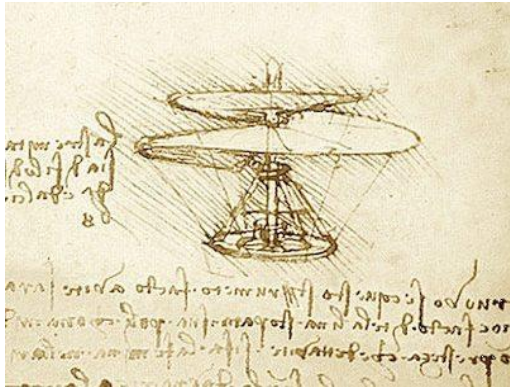


Figura 2 – Boceto del Tornillo Aéreo de Leonardo da Vinci [34]

En 1483, Leonardo Da Vinci diseñó una sofisticada aeronave capaz de la hazaña antes descrita. La máquina tenía un diámetro de 5 metros, recibía el nombre de *Tornillo Aéreo* y, con la adecuada fuerza conductora, se suponía capaz de rotar y elevarse. Sin embargo, dicha tecnología era irrealizable en la época.

A finales del siglo XIX las principales barreras a superar eran la carencia de estudios en el campo de la aerodinámica y la obtención de la energía necesaria para el despegue, pues la fuente de energía disponible, la máquina de vapor, tenía una relación peso-potencia demasiado elevada.

En la década de 1880, Thomas Edison experimentó en los Estados Unidos con diversas configuraciones para los rotores de aeronaves impulsadas por motores de *algodón de pólvora*¹. También investigó la posibilidad de emplear motores eléctricos.

Durante la Primera Guerra Mundial, el interés militar contribuyó al avance científico de la tecnología de vuelo en Gran Bretaña, Estados Unidos, Alemania, Francia e Italia.

Bajo el patrocinio de la armada estadounidense en 1922, George de Bothezat construyó una aeronave de 1700 kg, con un motor de 220 caballos y 4 propulsores de 6 aspas. Poseía además diversas hélices auxiliares para su control. Se trataba del primer aparato cuadrirrotor, pero nunca se elevó más de 5 metros sobre el suelo.

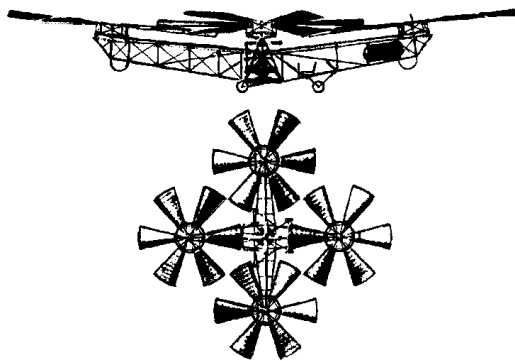


Figura 3 – Helicóptero de Bothezat [3]



Figura 4 – Autogiro Cierva C-40 [4]

Un año después, el ingeniero español Juan de la Cierva desarrolló el autogiro, antecesor directo del helicóptero. Era un vehículo mucho más ligero que el anterior y de construcción más simple, que logró realizar vuelos satisfactorios, pero carecía de la capacidad de despegar verticalmente o de mantenerse suspendido.

¹ Nitrato de celulosa [32],

Por fin, en 1939, Igor Sikorsky construyó el primer helicóptero, motivando el éxito de la aviación de ala giratoria y despegue vertical.

En la actualidad, la configuración de cuatro propulsores se utiliza principalmente en vehículos UAV de escala reducida. En este marco se han realizado enormes avances perfeccionando su control y estabilidad.

2.2.2. Los vehículos aéreos no tripulados

Fuentes [2], [5], [6].

El primer intento de vehículo aéreo guiado por radio fue el *"Aerial Target"*, desarrollado por A. M. Low en 1916, pero no fue hasta la Segunda Guerra Mundial cuando se realizaron verdaderos avances en dicho campo. Siendo los aviones a control remoto utilizados en misiones de ataque y para entrenar a los artilleros antiaéreos.

El desarrollo de los UAV fue propulsado por EE.UU. a partir de 1959 durante la guerra de Vietnam con el fin de minimizar la pérdida de pilotos americanos en territorio hostil. Se realizaron más de 3000 misiones militares bajo el programa de alto secreto de nombre *"Red Wagon"*.

Durante la guerra árabe-israelí de *Yom-Kipur* (1973) fue Israel quien finalmente desarrolló lo que se considera como el primer UAV moderno ante la imposibilidad de penetrar las defensas aéreas sirias. La adición de esta tecnología a la inteligencia israelí permitió un conocimiento mucho mayor del territorio enemigo y finalmente fue capaz de neutralizar por completo las defensas aéreas sirias en la guerra del Líbano (1982).

A partir de los años 80 y en la actualidad, el interés militar en los UAV continúa en crecimiento y estas inversiones potencian enormemente el avance de esta tecnología. Sin embargo, con la miniaturización de los sistemas, los vehículos no tripulados se están empleando también en los campos de la investigación científica, el ocio y el arte.

2.3. Técnicas de control para sistemas inestables

Pese a que hoy en día todavía existe gran diversidad de planteamientos en torno a la estructura física del quadrotor, sin duda el tópico que ha atraído la atención de un mayor número de publicaciones es el algoritmo de control, debido en gran medida a la naturaleza inestable² del sistema.

El control de sistemas dinámicos consiste en la obtención de acciones de control que permitan llevar y mantener unas salidas a una región delimitada mediante una estrategia de control [7].

En particular, el control de sistemas inestables es todavía hoy un área de gran importancia y continuo estudio.

² Sistema inestable: Proceso que, al recibir una entrada acotada, su salida no converge a un valor permanente.

Para contextualizar el trabajo realizado se mencionarán a continuación algunas técnicas pertenecientes a este campo [8], [9]:

- Control a través de la teoría de la estabilidad de Liapunov, método que se proclama capaz de asegurar la estabilidad asintótica del dron bajo ciertas condiciones.
- Técnicas de control PID.
- Técnicas de control adaptativo, que proporcionan buena respuesta con incertidumbres paramétricas y dinámicas sin modelar.
- Regulador Cuadrático Linear (LQR), que utiliza la información del estado completo y la ecuación de Ricatti para el cálculo de la acción de control. Computacionalmente resulta muy costoso.
- Métodos de control recursivo para sistemas no lineales. También resulta caro desde el punto de vista computacional.
- Control mediante feedback dinámico, para transformar el proceso en bucle cerrado en un sistema lineal, controlable y desacoplado.
- Técnicas de visión artificial y procesamiento de imagen.
- Técnicas de control difuso.
- Redes neuronales artificiales, que aplican la inteligencia artificial al campo del control de procesos.

2.4. Aplicaciones actuales de los drones

En los últimos años se ha producido un auge de proyectos que trabajan con mini-quadrotors de peso inferior a 10 kg. Reciben, entre otros, usos de tipo:

- Civil y comercial:
 - Científico, posibilitando el acceso y la exploración de lugares de otro modo inaccesibles, la cartografía detallada, la prospección de yacimientos, la gestión del medio ambiente, el control urbanístico, la hidrología, etc.
 - Con enfoque académico, ya que muchas instituciones investigadoras han desarrollado su propio sistema quadrotor con éxito. Cabe destacar los proyectos *Open-Source* recientes: Pixhawk, STARMAC, OS4 y X4-flyer. [10]
 - Logístico, como vehículo de carga y distribución de pequeñas mercancías. Como los Amazon Prime Air [11].
 - Artístico, sirviendo de inspiración para numerosas obras y mejorando la filmación de planos aéreos en películas actuales.
 - Orientado al ocio, en forma de juguetes de radio control. Este es el caso del Parrot AR.Drone que se maneja desde el smartphone.

- Militar:
 - En misiones de inteligencia, en misiones de reconocimiento de territorio hostil.
 - En combate, siendo ya frecuente que los UAV vayan armados.
 - Experimental, como sujetos de pruebas para el desarrollo de futuras tecnologías de aviación.
 - Como blanco para el entrenamiento de la artillería amiga o señuelo para confundir a la enemiga.



Figura 5 – Modelo de quadrotor Amazon Prime Air

2.5. Normativa

Desde el 30 de diciembre de 2017 la ley que regula el uso civil de aeronaves pilotadas por control remoto (RPAs) en España es el Real Decreto 1036/2017 [12]. En este apartado se hará referencia a las particularidades de dicha legislación que atañen a nuestra aeronave.

En cuanto a la responsabilidad del uso de RPAs:

“El piloto remoto será, en todo momento, el responsable de detectar y evitar posibles colisiones y otros peligros.” – Capítulo I, Artículo 4, RD 1036/2017.

Dado que todos los vuelos de prueba que se llevarán a cabo sucederán en el interior del laboratorio del Ai2, se hará referencia al Artículo 2 del RD 1036/2017, que estipula que dicha legislación no es de aplicación sobre “vuelos que se desarrollen en su integridad en espacios interiores completamente cerrados”. La AESA indica que en estos casos hay que seguir las recomendaciones básicas de seguridad y, por supuesto, contar con el permiso del titular del recinto.

2.6. Motivación y justificación del proyecto

En los últimos años está habiendo un gran interés por parte de la industria en el control de sistemas inestables. Esta es un área del conocimiento de la que se puede decir que todavía se encuentra en un estado de desarrollo relativamente temprano y el disponer de nociones actualizadas sobre este campo constituye hoy día una herramienta muy valiosa para el ingeniero.

En concreto, el control de aeronaves no tripuladas es un frente abierto en la ciencia de los sistemas inestables pues presenta retos como incertidumbres paramétricas, no linealidades, dinámicas entrelazadas y perturbaciones externas.

Existe ya en la literatura una amplia variedad de métodos aplicados al control de orientación de estos vehículos (ver apartado 2.3), sin embargo, las técnicas de control más extendidas continúan basándose en el tradicional algoritmo PID. Esto se debe principalmente a la necesidad de que el algoritmo empleado posea un bajo coste computacional que permita su ejecución a muy altas frecuencias en el ordenador de a bordo del vehículo.

Por este motivo se ha considerado importante realizar una comparativa de algoritmos de relativa sencillez y bajo coste de cálculo resaltando las ventajas e inconvenientes de las alternativas existentes en la industria actual.

Este trabajo tratará de realizar esto de formas a la vez didáctica y aplicada empleando un modelo de UAV quadrotor disponible en el laboratorio del departamento de automática.

Descripción de la Plataforma

A continuación, se procederá a describir el instrumental que se ha empleado en la realización de este trabajo de fin de grado. Se hará mención de cada uno de los componentes principales de la plataforma quadrotor y finalmente del banco de pruebas que nos ha permitido realizar experimentos sobre el dron con seguridad.

El material descrito es resultado de anteriores proyectos desarrollados en el laboratorio de Ai2, de cuyos resultados se parte para realizar el actual Trabajo de Fin de Grado.

Es importante destacar que, gracias a la modularidad de los sistemas empleados, estos serían fácilmente modificables para permitir su implementación en cualquier vehículo multirroto, o incluso en cualquier tipología de robot móvil.

3.1. El quadrotor



Figura 6 – El Quadrotor

El dron utilizado es el modelo que fue empleado en el TFM: [13]; ya que el montaje de un cuadricóptero no es objetivo del proyecto, centrándose este en su control.

Consta de una estructura rígida en forma de cruz en cuyos brazos se acoplan los propulsores. En su centro, sobre un esqueleto modular, se distribuye el chipset (ordenador de a bordo, driver de motores, IMU, receptor Wifi, etc.).

A continuación, se describirán de forma resumida sus elementos principales.

3.1.1. El ordenador de a bordo

El gobierno del quadrotor corre a cargo de una Raspberry Pi 2 [14], la cual ejecuta el sistema operativo Raspbian con el parche Preempt_RT³, que lo convierte en un SO en tiempo real. Esto es imprescindible para posibilitar su control, pues es necesario que las distintas tareas se ejecuten en un tiempo determinado [15].

La conexión con el equipo de tierra se realiza a través de una red Wifi *ad-hoc* entre ambos dispositivos.

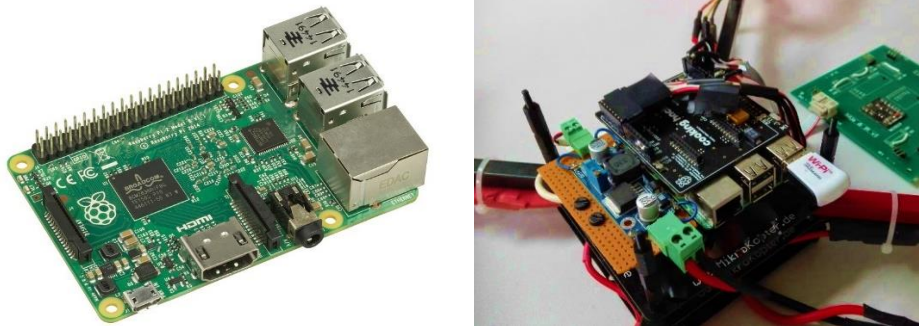


Figura 7 – Raspberry Pi 2 [14]

3.1.2. La IMU

La unidad de medición inercial (IMU) es un sistema de monitorizado imprescindible para el control de un cuadricóptero [16]. El chip consta de tres parejas de giroscopio-acelerómetro para medir la velocidad y aceleración en cada uno de los ángulos (*roll*, *pitch* y *yaw*) y de un magnetómetro que determina la orientación del vehículo respecto del campo magnético terrestre.

Para hacer posible la implementación de un control de orientación es necesaria una medida en tiempo real de cada uno de dichos ángulos. Sin embargo, no disponemos de una medida directa, sino que se ha de obtener a partir de la velocidad y aceleración proporcionadas por la IMU.

Esta se ha de calcular teniendo en cuenta las características de cada uno de los sensores:

³ *Preempt_RT* es un parche que actúa sobre el kernel de Raspbian permitiendo a la sintaxis estándar obtener características en tiempo real siempre y cuando se sigan ciertas pautas al crear las tareas [33].

El acelerómetro determina la aceleración en cada uno de los ejes y por tanto puede emplearse para obtener la orientación angular del vehículo empleando la proyección de la aceleración gravitatoria sobre la terna de ejes. Debido a la multitud de fuerzas que recibe el vehículo se trata de un sensor poco preciso en movimiento y la medida presentará mucho ruido.

El giroscopio obtiene la velocidad angular en cada eje con gran precisión. La magnitud del ángulo habrá de obtenerse integrando dicha medida y por este motivo presentará deriva con el tiempo.

El magnetómetro tiene la ventaja de que no presenta deriva a largo plazo y nos permitirá corregir la medida del giroscopio. Sin embargo, se ve afectado por los campos magnéticos de los equipos cercanos.

Se proponen dos maneras de lograr la fusión sensorial:

- A través de un filtro complementario [17]:

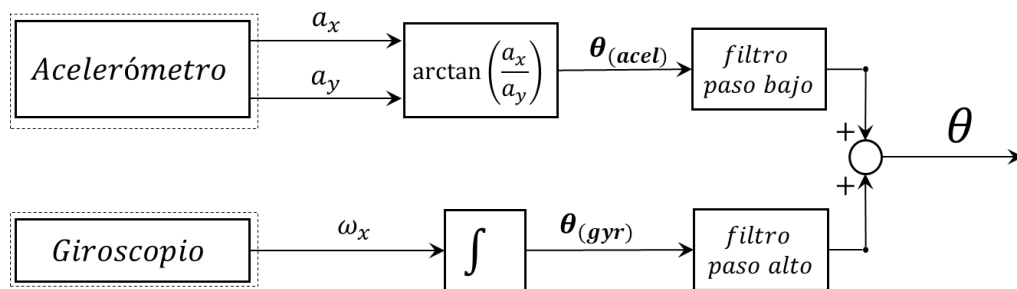


Figura 8 – Esquema del filtro complementario

Este sistema se basa en el hecho de que, mientras el giroscopio nos permite obtener la medida angular en cada eje con precisión a corto plazo, pero con deriva en el futuro; el acelerómetro y el magnetómetro nos proporcionan una medida ruidosa, pero sin deriva.

Para obtener una medición fiable y estable en todo momento se eliminarán los errores de baja frecuencia que se integran de la medida del giroscopio mediante un filtro paso-bajo y se eliminará el ruido de medida del acelerómetro usando un filtro paso-alto para, posteriormente, combinar ambas medidas.

Para obtener la magnitud de los ángulos *roll* y *pitch* se combinarán las salidas de acelerómetro y giroscopio en dichos ejes. Para el *yaw*, por otra parte, se emplearán giroscopio y magnetómetro.

- Empleando un filtro de Kalman con el ajuste adecuado:

Considerado como el estimador óptimo para sistemas dinámicos lineales, el filtro de Kalman es un algoritmo recursivo que permite obtener predicciones del sistema siempre y cuando las covarianzas de los errores de medida se puedan asumir de distribución Gaussiana, estacionaria con el tiempo y de media cero [18].

Este sistema funciona en dos fases:

1. Fase de predicción, en la cual se usa el modelo dinámico del sistema y la covarianza del error del proceso para realizar una predicción del estado del proceso.
2. Actualización de la medida empleando las mediciones combinadas de los sensores y la covarianza de su error. Dicha medición se compara con la predicción anterior y de este modo se ajusta el algoritmo para el siguiente ciclo.

El dron con que trabajamos lleva implementado un filtro de Kalman extendido ajustado experimentalmente. La comunicación con la IMU se lleva a cabo mediante el protocolo i2c a 450k baudios [13].

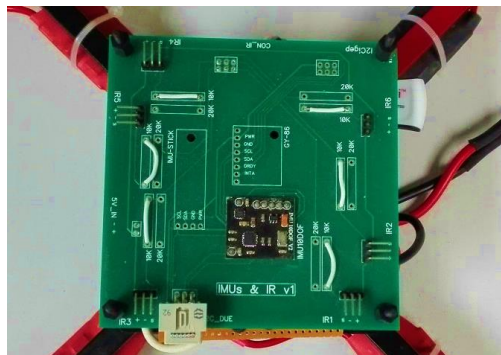


Figura 9 - La IMU

3.1.3. Los actuadores

Se dispone de cuatro propulsores (hélices) que generan los momentos requeridos como acción de control del sistema. Estos son movidos por cuatro motores sin escobillas anclados a los cuatro brazos del dron que permiten grandes velocidades de giro.

Para generar las tensiones necesarias para su funcionamiento y transmitirles las referencias de velocidad se dispone de drivers que emplean señales PWM de frecuencia máxima 450Hz. Su rango de actuación se encuentra fijado de 0 a 1 [6].

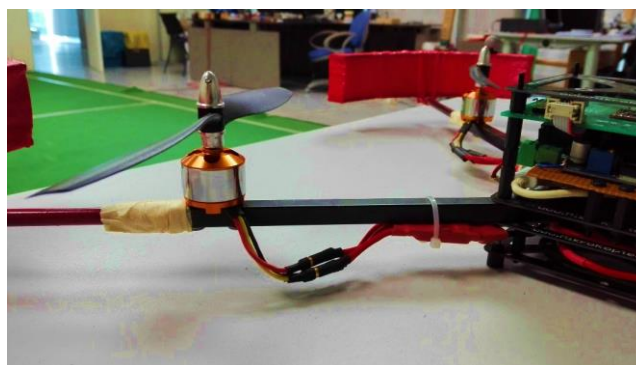


Figura 10 - Motor del dron (detalle)

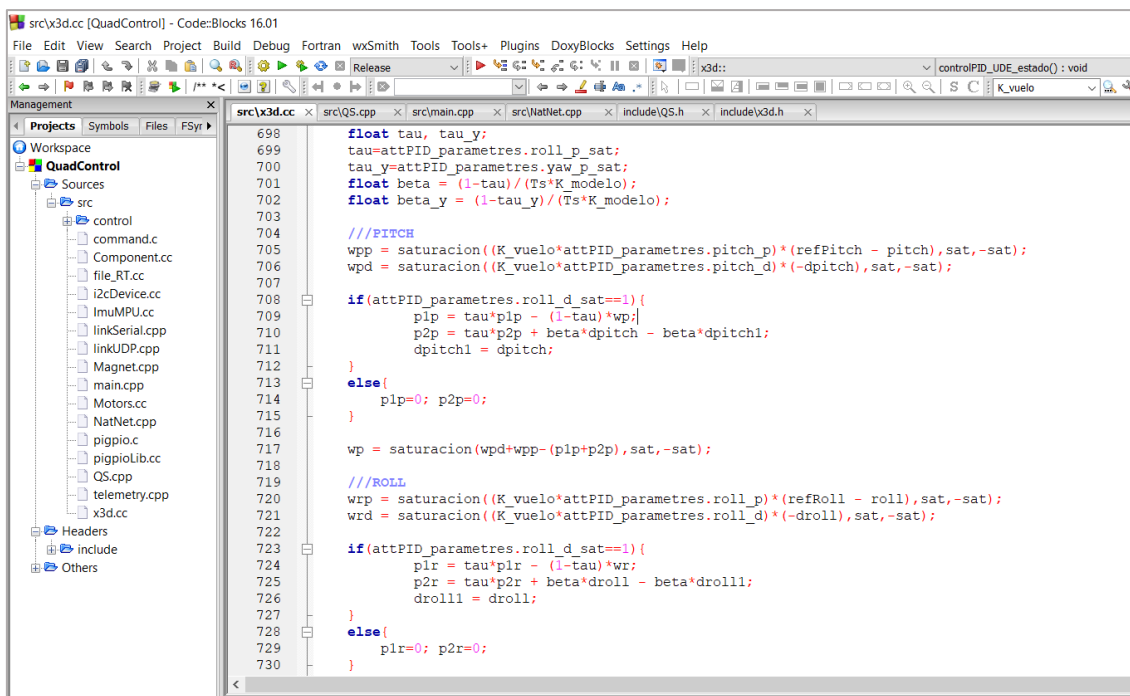
3.1.4. El código

El software del quadrotor se encuentra programado en lenguaje C++ (programación orientada a objetos). Se trata de un lenguaje de bajo nivel que posibilita la ejecución del programa en tiempo real.

El programa *Quadcontrol* se divide por clases según su función dentro del sistema (contacto con el ordenador de tierra, lectura de cada uno de los sensores, fusión sensorial, señal de los motores, algoritmo de control, etc.).

En este proyecto se modificará la clase *x3d.cc*, que contiene el algoritmo de control, para implementar las distintas técnicas y experimentar con el vehículo. Esta trabaja con un tiempo de muestreo de 4ms.

Para escribir y compilar el código se ha empleado el programa *open-source* Code::Blocks [19].



```
698     float tau, tau_y;
699     tau=attPID_parametres.roll_p_sat;
700     tau_y=attPID_parametres.yaw_p_sat;
701     float beta = (1-tau)/(Ts*K_modelo);
702     float beta_y = (1-tau_y)/(Ts*K_modelo);
703
704     //PITCH
705     wpp = saturacion((K_vuelo*attPID_parametres.pitch_p)*(refPitch - pitch), sat, -sat);
706     wpd = saturacion((K_vuelo*attPID_parametres.pitch_d)*(-dpitch), sat, -sat);
707
708     if(attPID_parametres.roll_d_sat==1){
709         p1p = tau*p1p - (1-tau)*wp;
710         p2p = tau*p2p + beta*dpitch - beta*dpitch1;
711         dpitch1 = dpitch;
712     }
713     else{
714         p1p=0; p2p=0;
715     }
716
717     wp = saturacion(wpd+wpp-(p1p+p2p), sat, -sat);
718
719     //ROLL
720     wrp = saturacion((K_vuelo*attPID_parametres.roll_p)*(refRoll - roll), sat, -sat);
721     wrd = saturacion((K_vuelo*attPID_parametres.roll_d)*(-droll), sat, -sat);
722
723     if(attPID_parametres.roll_d_sat==1){
724         p1r = tau*p1r - (1-tau)*wr;
725         p2r = tau*p2r + beta*droll - beta*droll1;
726         droll1 = droll;
727     }
728     else{
729         p1r=0; p2r=0;
730     }
731
```

Figura 11 – Proyecto QuadControl en el programa Code::Blocks

3.2. La fuente de alimentación

Nuestro sistema se alimenta a 12V, el estándar de alimentación utilizada en quadrotors de este calibre. Durante los ensayos en el laboratorio se empleará para ello un convertidor AC/DC, evitando de este modo depender de los ciclos de carga/descarga de una batería. Por supuesto, en caso de necesidad sería posible añadir un piso más a la estructura central permitiendo la sujeción de una batería.

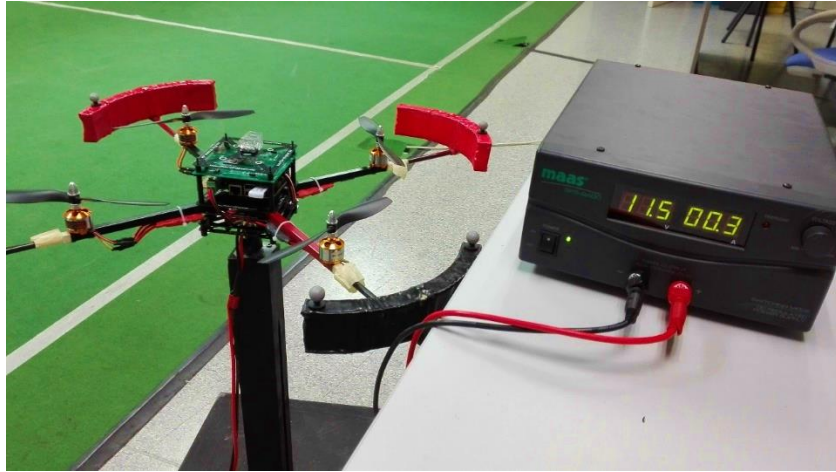


Figura 12 – Fuente de alimentación

3.3. El ordenador de tierra

El equipo empleado es un PC con sistema operativo Windows 10. La conexión con la Raspberry Pi se realiza, como ya se ha mencionado anteriormente, a través de una red *ad-hoc* inalámbrica que proporciona un canal de comunicación UDP para los datos, cifrados mediante el protocolo Mavlink [13].

Se dispone de acceso a la consola de la Raspberry Pi a través del *PuTTY* [20], un cliente SSH y telnet de licencia libre, y acceso a su explorador de archivos empleando *WinSCP* [21], un cliente SFTP, SCP y FTP también gratuito. El uso combinado de ambas aplicaciones permite la transmisión y ejecución de los programas en el ordenador de a bordo desde el PC base.

Por último, la interfaz hombre máquina (HMI) nos permite, durante la ejecución del programa *Quadcontrol* en la Raspberry, enviarle instrucciones en tiempo real como, por ejemplo: parámetros de control, referencias, señal de start/stop, etc. Este programa fue desarrollado en lenguaje Java para poder ejecutarse en cualquier sistema operativo [13].

3.4. El banco de ensayos

Para realizar ensayos de forma segura se dispone de una plataforma consistente de un poste de madera anclado a una base. En su extremo superior una rótula metálica nos permite anclar el quadrotor permitiendo 3 grados de libertad (los tres ángulos de giro).

Los ensayos en sobre el vehículo anclado a la rótula, sin embargo, tienen la desventaja de que alteran de forma apreciable el comportamiento del sistema (Ver apartado 5.2). Aun así, las ventajas en seguridad y manejabilidad que nos aporta el banco de ensayos superan este inconveniente con creces.

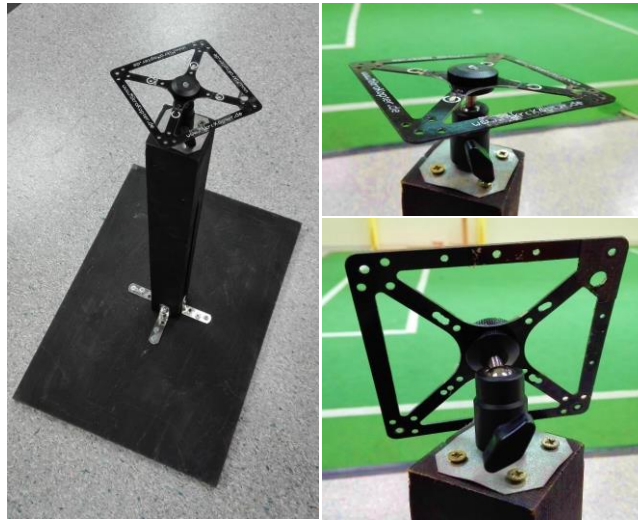


Figura 13 – Banco de ensayos o r tula

3.5. El sistema *Optitrack*

El sistema *Optitrack* [22] consiste en de una instalaci n de captura de movimiento, desarrollada por la compa a del mismo nombre, que consiste en un equipo de 6 c maras distribuidas formando un rect ngulo, un procesador independiente de im genes para la visi n artificial y el software de la empresa, *Motive*, que se ejecuta en el PC.

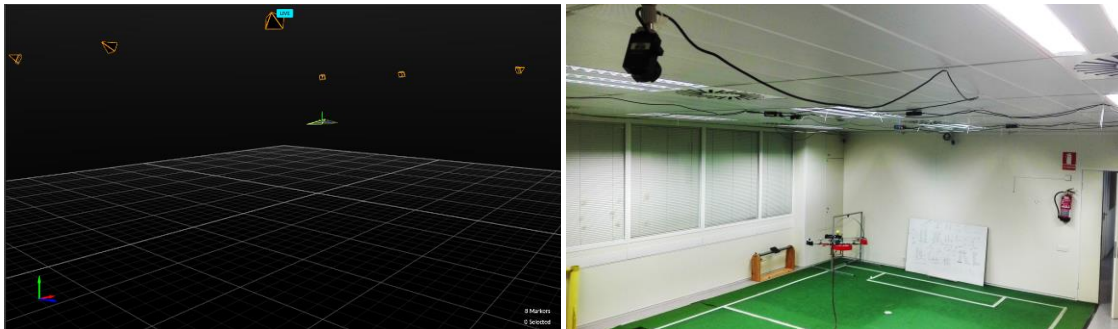


Figura 14 – Reconstrucci n de la posici n del dron (dcha.) en el programa *Motive* (izqda.)

Este instrumental se emplea usualmente para t cnicas de captura de movimiento en cine y videojuegos, pero es muy conveniente para el control de aeronaves en el interior de edificios, donde la se al GPS es de peor calidad.

Las c maras son dispositivos de emisi n y recepci n de luz infrarroja. Dicha radiaci n se refleja en ocho marcadores ubicados en puntos estrat gicos del veh culo permitiendo su localizaci n con exactitud milim trica en sus 6 grados de libertad.

El sistema del que se dispone en el laboratorio no es de  ltima generaci n, pues sus componentes se encuentran ya descatalogados por la empresa [23]. Sin embargo, bastar n

para obtener una medida fiable, superior a la que proporcionaría un GPS, de la ubicación del vehículo.

Nuestro equipo trabaja a una frecuencia máxima de 100Hz (10ms) y las medidas son enviadas al ordenador de tierra y desde ahí a la Raspberry Pi.



Figura 15 – Cámara Flex:V100 del equipo Optitrack del laboratorio (detalle)

Modelado del sistema

4.1. Funcionamiento del quadrotor

La aeronave posee una configuración de cuatro motores coplanarios, con los cuales es capaz de realizar cualquier maniobra en el espacio. Se trata por tanto de un sistema subactuado, pues posee cuatro entradas (velocidades de los motores) y con ellas se han de controlar los seis grados de libertad del vehículo (ver Figura 16). Esto se logra mediante el control diferencial de las velocidades de los propulsores.

Como ya se ha expuesto en el capítulo anterior, es imprescindible que los rotores 1 y 3 giren en sentido horario y los rotores 2 y 4 en antihorario de modo que se cancele el par resultante en el centro del aparato. En caso de que se desee generar rotación en el eje del *yaw* (ψ) se eliminará el equilibrio antes mencionado acelerando una de las parejas de propulsores y decelerando la otra.

La rotación en los ejes del *roll* (φ) y el *pitch* (θ) se realiza mediante una acción diferencial sobre la pareja de motores ubicados sobre el eje complementario. En definitiva, si buscamos generar una rotación en sentido horario sobre el eje φ , habremos de aumentar f_2 y disminuir f_4 .

Por último, para lograr desplazamientos en los ejes x e y habremos de inclinar el aparato sobre θ y φ respectivamente.

Esta explicación corresponde a un quadrotor en estructura de cruz (+), en la cual los ejes de coordenadas relativas al vehículo se encuentran alineados con los brazos de este. Esta configuración se ha elegido para la explicación teórica por ser más didáctica e intuitiva. Para más información ver el subapartado 4.4.

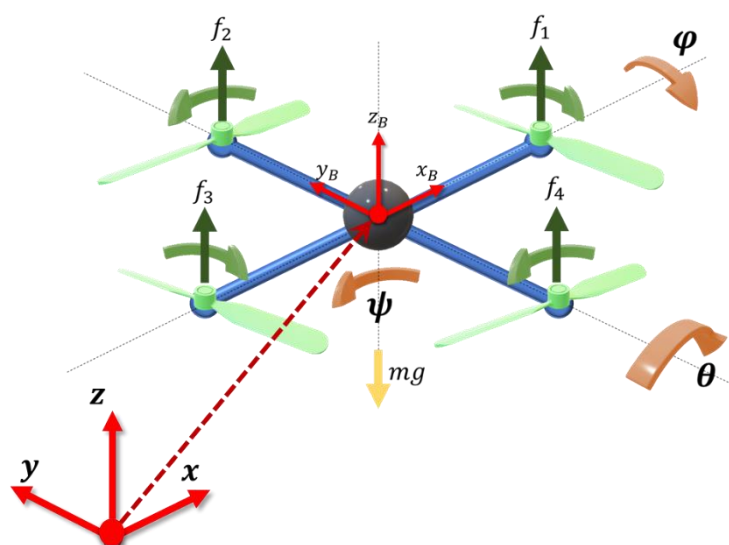


Figura 16 – Esquema ilustrativo del quadrotor en cruz.

4.2. Modelo Teórico del quadrotor

Fuentes: [24], [25]

Para el cálculo del modelo del dron en este proyecto se utilizará la formulación de Euler-Lagrange.

Además, buscando facilitar el cálculo, se tomarán las siguientes hipótesis simplificativas:

- Se considerará el UAV como un sólido rígido.
- Se despreciará el efecto suelo.
- La estructura y distribución de masas del dron se considerará simétrica.
- Supondremos que el centro de masas es coincidente con el origen del sistema de coordenadas que tomaremos para el quadrotor.
- Se supondrá que los motores son iguales, con dinámica despreciable respecto de la del vehículo.
- Se tomará el quadrotor en posición de cruz.

De este modo el modelo constará de una matriz de inercia diagonal.

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

Posición del sólido rígido:

Para expresar la orientación del cuerpo se utilizará la convención -XYZ (giros sucesivos en torno a los ejes x , y' , z''). Por tanto, se emplearán los ángulos de Tait-Bryan como coordenadas generalizadas de orientación:

$$\eta = \begin{bmatrix} \widehat{yz} \\ \widehat{xz} \\ \widehat{xy} \end{bmatrix} = \begin{bmatrix} \varphi \\ \theta \\ \psi \end{bmatrix} \quad (2)$$

Además, consideraremos dos sistemas cartesianos: el sistema ligado al cuerpo del dron (coordenadas relativas) y el sistema inercial, solidario a la tierra (coordenadas absolutas). Este último será nuestro vector de coordenadas generalizadas de la posición.

Respectivamente:

$$\xi_B = \begin{bmatrix} x_B \\ y_B \\ z_B \end{bmatrix} \quad y \quad \xi = \begin{bmatrix} x \\ y \\ z \end{bmatrix} \quad (3)$$

La matriz de rotación del cuerpo se obtiene mediante el producto de las tres matrices de rotación correspondientes a dichos ángulos.

$$\mathbb{R} = R\psi \cdot R\theta \cdot R\varphi$$

$$\mathbb{R} = \begin{bmatrix} \cos \psi & -\sin \psi & 0 \\ \sin \psi & \cos \psi & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos \theta & 0 & \sin \theta \\ 0 & 1 & 0 \\ -\sin \theta & 0 & \cos \theta \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \varphi & -\sin \varphi \\ 0 & \sin \varphi & \cos \varphi \end{bmatrix}$$

$$\mathbb{R} = \begin{bmatrix} \cos \psi \cdot \cos \theta & \cos \psi \cdot \sin \theta \cdot \sin \varphi - \sin \psi \cdot \cos \varphi & \cos \psi \cdot \sin \theta \cdot \cos \varphi + \sin \psi \cdot \sin \varphi \\ \sin \psi \cdot \cos \theta & \sin \psi \cdot \sin \theta \cdot \sin \varphi + \cos \psi \cdot \cos \varphi & \sin \psi \cdot \sin \theta \cdot \cos \varphi - \cos \psi \cdot \sin \varphi \\ -\sin \theta & \cos \theta \cdot \sin \varphi & \cos \theta \cdot \cos \varphi \end{bmatrix} \quad (4)$$

Operando con dicha matriz trasladaremos las medidas del sistema de coordenadas relativo al cuerpo al sistema inercial:

$$\xi = \mathbb{R} \cdot \xi_B \quad (5)$$

Velocidad del sólido rígido:

A través de la IMU instalada en el dron obtendremos mediciones de la velocidad angular del vehículo (Ω) en el sistema de coordenadas relativas a dicho cuerpo rígido.

Dicha velocidad angular se relaciona con las derivadas de los ángulos de Tait-Bryan a través de la matriz W :

$$\Omega = W(\eta) \dot{\eta} \quad (6)$$

Donde

$$W(\eta) = \begin{bmatrix} 1 & 0 & \sin \theta \\ 0 & \cos \varphi & \sin \varphi \cos \theta \\ 0 & -\sin \varphi & \cos \varphi \cos \theta \end{bmatrix} \quad (6.1)$$

Propuesta de Euler-Lagrange:

Las ecuaciones de la dinámica del quadrotor se pueden expresar mediante la formulación de Euler-Lagrange. Esta toma como base los conceptos de energía cinética y potencial.

$$\frac{d}{dt} \left[\frac{\partial L}{\partial \dot{q}} \right] - \frac{\partial L}{\partial q} = Q \quad (7)$$

Donde

- Q son las fuerzas o pares generalizados:

$$Q = \begin{bmatrix} F_{\xi} \\ \tau \end{bmatrix} \quad (8)$$

Siendo:

$$F_{\xi} = \mathbb{R} F \quad \text{tal que} \quad F = \begin{bmatrix} 0 \\ 0 \\ \sum \text{fuerza motores} \end{bmatrix} \quad (8.1)$$

$$\tau = \begin{bmatrix} \tau_{\varphi} \\ \tau_{\theta} \\ \tau_{\psi} \end{bmatrix} = \begin{bmatrix} (f_{\text{motor } 2} - f_{\text{motor } 4})l \\ (f_{\text{motor } 1} - f_{\text{motor } 3})l \\ \sum \tau_{\text{motores}} \end{bmatrix} \quad (8.2)$$

- q son las coordenadas generalizadas.

$$q = \begin{bmatrix} \xi \\ \eta \end{bmatrix} = \begin{bmatrix} x & y & z \\ \varphi & \theta & \psi \end{bmatrix} \quad (9)$$

- \dot{q} es la primera derivada respecto al tiempo de las coordenadas generalizadas.
- L es la función lagrangiana; resta de las energías cinética y potencial del sistema:

$$L = E_c - E_p$$

$$L = (E_{\text{ctraslación}} + E_{\text{crotación}}) - E_p$$

$E_{\text{ctraslación}} = \frac{m}{2} \dot{\xi}^T \dot{\xi}$, siendo m la masa del cuerpo y $\dot{\xi}$ la velocidad lineal del sistema expresada en el sistema inercial.

$E_{\text{crotación}} = \frac{1}{2} \Omega^T I \Omega$, siendo I su matriz de inercia y Ω la velocidad angular del vehículo en el sistema cuerpo. Trasladando dichas velocidades al sistema inercial:

$$E_{\text{crotación}} = \frac{1}{2} \Omega^T I \Omega = \frac{1}{2} (W \dot{\eta})^T I (W \dot{\eta}) = \frac{1}{2} \dot{\eta}^T \mathbb{J} \dot{\eta} \quad \text{tal que} \quad \mathbb{J} = W^T I W$$

$$\mathbb{J} = \begin{bmatrix} I_{xx} & 0 & -I_{xx} \sin \theta \\ 0 & I_{yy} \cos^2 \varphi + I_{zz} \sin^2 \varphi & (I_{yy} - I_{zz}) \cos \varphi \sin \varphi \cos \theta \\ -I_{xx} \sin \theta & (I_{yy} - I_{zz}) \cos \varphi \sin \varphi \cos \theta & I_{xx} \sin^2 \theta + I_{yy} \sin^2 \varphi \cos^2 \theta + I_{zz} \cos^2 \varphi \cos^2 \theta \end{bmatrix}$$

$E_p = mgz$, siendo m la masa del cuerpo, z la coordenada generalizada $\xi(3)$ (la altura a la que se encuentra el cuerpo) y g es la aceleración provocada por la gravedad terrestre ($g = 9,81 \text{ m/s}^2$).

Por tanto:

$$L = \left(\frac{m}{2} \dot{\xi}^T \dot{\xi} + \frac{1}{2} \dot{\eta}^T \mathbb{J} \dot{\eta} \right) - mgz \quad (10)$$

En consecuencia, podremos dividir dicha expresión en dos partes:

El lagrangiano traslacional: $L_{trans} = \frac{m}{2} \dot{\xi}^T \dot{\xi} - mgz$;

Y el lagrangiano rotacional: $L_{rot} = E_{rotación} = \frac{1}{2} \dot{\eta}^T \mathbb{J} \dot{\eta}$

La ecuación de Euler-Lagrange se podrá a su vez fraccionar de la misma forma en base a dichas consideraciones.

Ecuación del modelo traslacional:

$$\frac{d}{dt} \left[\frac{\partial L_{trans}}{\partial \dot{\xi}} \right] - \frac{\partial L_{trans}}{\partial \xi} = F_{\xi}$$

$$m\ddot{\xi} + \begin{pmatrix} 0 \\ 0 \\ mg \end{pmatrix} = F_{\xi} \quad (11)$$

Si desarrollamos:

$$\begin{aligned} \ddot{x} &= \frac{1}{m} \left[(\cos \psi \sin \theta \cos \varphi + \sin \psi \sin \varphi) \cdot \sum_{i=0}^4 f_{mot}^{(i)} + F_{aerodinámica}^{(x)} \right] \\ \ddot{y} &= \frac{1}{m} \left[(\sin \psi \sin \theta \cos \varphi - \cos \psi \sin \varphi) \cdot \sum_{i=0}^4 f_{mot}^{(i)} + F_{aerodinámica}^{(y)} \right] \\ \ddot{z} &= -g + \frac{1}{m} \left[(\cos \theta \cos \varphi) \cdot \sum_{i=0}^4 f_{mot}^{(i)} + F_{aerodinámica}^{(z)} \right] \end{aligned} \quad (12)$$

Ecuación del modelo rotacional:

$$\frac{d}{dt} \left[\frac{\partial L_{rot}}{\partial \dot{\eta}} \right] - \frac{\partial L_{rot}}{\partial \eta} = \tau$$

$$\mathbb{J} \ddot{\eta} + \left[\dot{\mathbb{J}} - \frac{1}{2} \frac{\partial}{\partial \eta} (\dot{\eta}^T \mathbb{J}) \right] \dot{\eta} = \tau$$

$$\mathbb{J} \ddot{\eta} + [C(\eta, \dot{\eta})] \dot{\eta} = \tau$$

Despejando:

$$\ddot{\eta} = \mathbb{J}^{-1} [\tau - [C(\eta, \dot{\eta})] \dot{\eta}] \quad (13)$$

$C(\eta, \dot{\eta})$ hace referencia a los fenómenos giroscópicos y centrífugos y es comúnmente llamado término de Coriolis. Tiene la forma:

$$C(\eta, \dot{\eta}) = \begin{bmatrix} c_{11} & c_{12} & c_{13} \\ c_{21} & c_{22} & c_{23} \\ c_{31} & c_{32} & c_{33} \end{bmatrix} \quad (13.1)$$

$$c_{11} = 0$$

$$c_{12} = (I_{yy} - I_{zz})(\dot{\theta} \cos \varphi \sin \varphi + \dot{\psi} \sin^2 \varphi \cos \theta) + (I_{zz} - I_{yy})\dot{\psi} \cos^2 \varphi \cos \theta - I_{xx} \dot{\psi} \cos \theta$$

$$c_{13} = (I_{zz} - I_{yy})\dot{\psi} \cos \varphi \sin \varphi \cos^2 \theta$$

$$c_{21} = (I_{zz} - I_{yy})(\dot{\theta} \cos \varphi \sin \varphi + \dot{\psi} \sin^2 \varphi \cos \theta) + (I_{yy} - I_{zz})\dot{\psi} \cos^2 \varphi \cos \theta + I_{xx} \dot{\psi} \cos \theta$$

$$c_{22} = (I_{zz} - I_{yy}) \dot{\varphi} \cos \varphi \sin \varphi$$

$$c_{23} = -I_{xx} \dot{\psi} \sin \theta \cos \theta + I_{yy} \dot{\psi} \sin^2 \varphi \cos \theta \sin \theta + I_{zz} \dot{\psi} \cos^2 \varphi \sin \theta \cos \theta$$

$$c_{31} = (I_{yy} - I_{zz})\dot{\psi} \cos^2 \theta \sin \varphi \cos \varphi - I_{zz} \dot{\theta} \cos \theta$$

$$c_{32} = (I_{zz} - I_{yy})(\dot{\theta} \cos \varphi \sin \varphi \sin \theta + \dot{\varphi} \sin^2 \varphi \cos \theta) + (I_{yy} - I_{zz}) \dot{\varphi} \cos^2 \varphi \cos \theta + I_{xx} \dot{\psi} \sin \varphi \cos \theta - I_{yy} \dot{\psi} \sin^2 \varphi \sin \theta \cos \theta - I_{zz} \dot{\psi} \cos^2 \varphi \sin \theta \cos \theta$$

$$c_{33} = (I_{yy} - I_{zz}) \dot{\varphi} \cos \varphi \sin \varphi \cos^2 \theta - I_{xx} \dot{\theta} \sin^2 \varphi \cos \theta \sin \theta - I_{zz} \dot{\theta} \cos^2 \varphi \cos \theta \sin \theta + I_{xx} \dot{\theta} \cos \theta \sin \theta$$

Este modelo matemático evidencia que es posible fragmentar nuestro sistema en dos subsistemas que se solapan el uno sobre el otro:

1. El subsistema de orientación; cuyas entradas son los pares creados por los cuatro motores (τ) y cuya salida es la orientación del vehículo (ángulos de Tait-Bryan).
2. El subsistema de traslación; cuyas entradas son la orientación del vehículo (salida del subsistema 1) y el empuje de los motores (F), y cuya salida es la posición del vehículo (x, y, z).

4.3. Modelo lineal aproximado:

Para el diseño del control de orientación se evidencia necesaria la obtención de un modelo lineal más sencillo con el que se trabajará a partir de este capítulo.

Se considerará como punto de operación el origen de coordenadas (lugar de estabilización del quadrotor). Por tanto, se podrá considerar la aproximación de pequeño ángulo para el pitch y el roll.

$$\varphi = \theta = 0$$

Aplicando esto en el modelo rotacional;

$$\mathbb{J} \dot{\eta} = \tau - [C(\eta, \dot{\eta})]\dot{\eta}$$

Obtendremos:

$$I_{xx} \ddot{\varphi} = \tau_{\varphi} - (I_{zz} - I_{yy} - I_{xx})\dot{\psi}\dot{\theta}$$

$$I_{yy} \ddot{\theta} = \tau_{\theta} - (I_{yy} - I_{zz} + I_{xx})\dot{\varphi}\dot{\psi}$$

$$I_{zz} \ddot{\psi} = \tau_{\psi} - (I_{yy} - I_{zz} - I_{xx})\dot{\varphi}\dot{\theta}$$

Además, supondremos despreciable el término de Coriolis respecto del par creado por los motores debido a la hipótesis de pequeños ángulos y lo consideraremos como una perturbación para nuestro control:

$$\begin{aligned} \ddot{\varphi} &\simeq \frac{\tau_{\varphi}}{I_{xx}} + f(\dot{\varphi}, \dot{\theta}, \dot{\psi}, d_{externa}) \\ \ddot{\theta} &\simeq \frac{\tau_{\theta}}{I_{yy}} + f(\dot{\varphi}, \dot{\theta}, \dot{\psi}, d_{externa}) \\ \ddot{\psi} &\simeq \frac{\tau_{\psi}}{I_{zz}} + f(\dot{\varphi}, \dot{\theta}, \dot{\psi}, d_{externa}) \end{aligned} \tag{14}$$

Aplicando la transformada de Laplace con condiciones iniciales nulas para resolver las ecuaciones se desprende que trabajaremos con un doble integrador como modelo aproximado del quadrotor para el diseño en el *lugar de las raíces*:

$$s^2\varphi = K \cdot \tau_\varphi + f; \quad \varphi = \frac{K}{s^2} \cdot \tau_\varphi + \frac{1}{s^2} \cdot f \quad \longrightarrow \quad y = \frac{K}{s^2} \cdot u + \frac{1}{s^2} \cdot pert \quad (15)$$

Expresando este modelo en *espacio de estados*:

$$\begin{cases} \begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 0 \\ K \end{bmatrix} u + pert \\ \begin{bmatrix} y_1 \\ y_2 \end{bmatrix} = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \quad (\text{salida}) \end{cases}$$

O lo que es lo mismo:

$$\begin{cases} \dot{x} = Ax + Bu + pert \\ y = x \end{cases} \quad (16)$$

4.4. Quadrotor en *cruz* y en *equis*

En función de cómo ubiquemos los ejes de nuestro sistema de coordenadas podemos realizar el control de nuestro vehículo en *cruz* o en *equis*. Como se ha explicado antes de forma breve, en el control en *cruz* los ejes x e y del sistema de coordenadas relativas al vehículo coinciden con sus brazos. En cambio, en el control en *equis* estos se encuentran desfasados 45° (Figura 17).

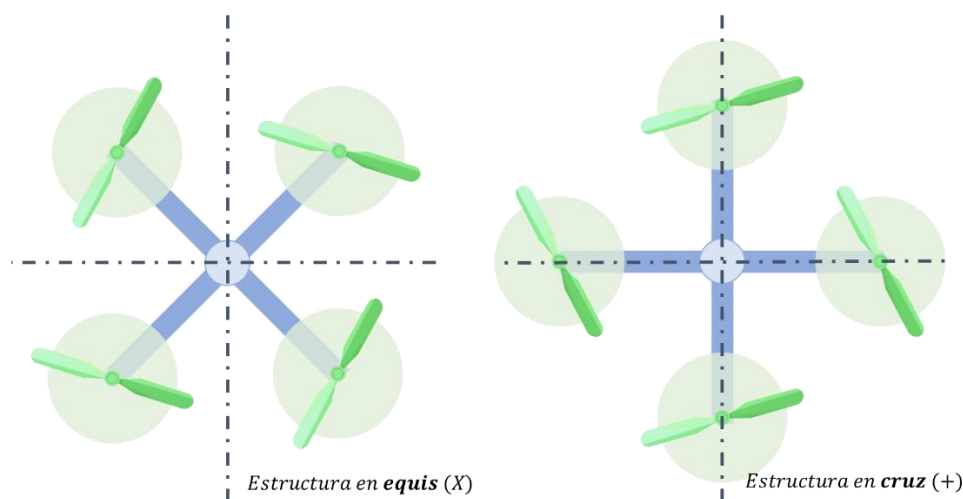


Figura 17 - Estructuras de control de un quadricóptero

Esta consideración no modifica el modelo del quadrotor, únicamente se ven modificadas las ecuaciones de control de los propulsores.

El control en *cruz* se emplea principalmente con fines didácticos por su fácil entendimiento. Cada rotación en torno a un eje es producto únicamente de la acción complementaria de dos motores simétricos, lo cual simplifica el esquema.

En el control en *equis*, sin embargo, intervienen los cuatro propulsores en todos los giros debido a que estos son simétricos dos a dos respecto a ambos ejes. Esto reduce la potencia requerida individualmente en cada motor para maniobrar con el vehículo y previene la saturación de la acción de control. Además, lograremos obtener una respuesta más rápida.

En los ensayos sobre el banco de trabajo se ha empleado control en *cruz*, sin embargo, si se deseara utilizar control en *equis* para el vuelo libre tan solo habrían de rotarse los ejes del sistema de coordenadas de la IMU 45° y modificar las ecuaciones de salida de los motores.

Identificación

Para hacer posible el diseño de un control de orientación adecuado a nuestro hardware será esencial un conocimiento más profundo de sus características. Nos basaremos para ello en las expresiones obtenidas en el capítulo previo y, mediante ensayos sobre el proceso real, obtendremos el modelo particular del dron.

Dicho modelo podrá ser utilizado como base para futuros diseños y simulaciones.

Como se aprecia en el modelo dinámico del proceso (apartado 4.2), la orientación del quadrotor está íntegramente relacionada con su traslación. Por este motivo será imprescindible el anclaje del dron al banco de trabajo para la ejecución de los ensayos. Dicho banco nos permitirá acceder de forma aislada al subsistema rotacional bloqueando los tres grados de libertad restantes y permitiendo su estudio.

5.1. Método empleado:

Al tratarse nuestro modelo de un sistema inestable no se podrá hacer uso de los métodos habituales de identificación en bucle abierto y será necesario aplicar un control previo. Dicho control habrá de realizarse a una alta frecuencia ($T_s=4ms$) debido a la rápida dinámica del sistema.

El proceso que se va a seguir es el siguiente:

Para obtener un sistema estable se implementará un bucle de control similar a un PD en el código del dron. Suponiendo que la estructura del modelo del quadrotor se ajusta a un doble integrador, la función de transferencia obtenida en bucle cerrado será la siguiente:

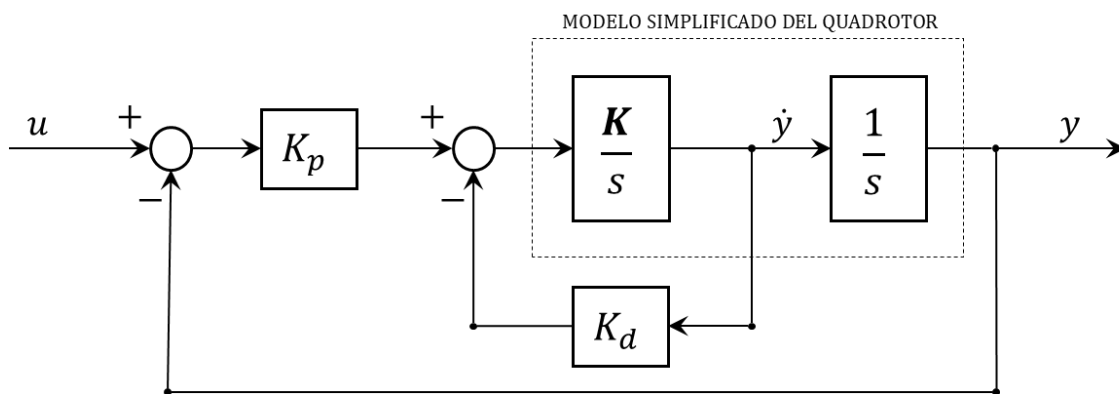


Figura 18 – Bucle de control para la identificación del sistema quadrotor

$$G_{BC} = \frac{y}{u} = \frac{K_p \mathbf{K}}{s(s + K_d \mathbf{K}) + K_p \mathbf{K}} = \frac{1}{\frac{1}{K_p \mathbf{K}} s^2 + \frac{K_d}{K_p} s + 1} = \frac{1}{as^2 + bs + 1}$$

Se realizará un ajuste inicial empírico de los parámetros del control (K_p, K_d) buscando que el vehículo se mantenga estable sobre la rótula.

Sobre este sistema, ya estable en bucle cerrado, se aplicarán escalones sucesivos en torno al punto de funcionamiento (origen de coordenadas) y se buscará el valor de la constante proporcional de nuestro modelo que mejor se ajuste al comportamiento obtenido en los ensayos.

Las medidas angulares arrojadas por la IMU utilizan como unidades los grados ($^{\circ}$). Consideraremos para esta identificación que las constantes de los modelos en los ángulos perpendiculares de pitch y roll serán aproximadamente iguales.

Para el procesado de los datos obtenidos en los ensayos se utilizará la herramienta System Identification Toolbox™ de MATLAB®.

5.2. Identificación de la orientación en *pitch* y *roll*

Dado que suponemos que nuestro sistema es simétrico, consideraremos equivalentes las funciones de transferencias de los procesos de balanceo y cabeceo y realizaremos una misma identificación para ambos.

El controlador empleado en los ensayos tiene como parámetros $K_p = 0,009$ y $K_d = 0,007$. Por tanto, según el modelo propuesto, la función de transferencia en bucle cerrado será:

$$G_{BC} = \frac{1}{\frac{1}{0,009 \cdot \mathbf{K}} s^2 + 0,78s + 1} \quad (17)$$

Tras 10 ensayos, se desprende que la función de transferencia que mejor se ajusta a todos ellos es

$$G_{EXP} = \frac{2}{a \cdot s^2 + 2s + 1} \quad (18)$$

Siendo $a \in [0,14; 1,13]$ y $\bar{a} = 0,23$; por lo que $K \in [100; 800]$ y $\bar{K} \simeq 480$.

Se observa que el modelo del doble integrador no se ajusta bien a los ensayos en la rótula: La constante K varía considerablemente de un ensayo a otro, ganancia del sistema no es unitaria y el término b obtenido en los ensayos no se ajusta al esperado. Esto se debe a que varias consideraciones que se han tomado para el modelado del dron en vuelo libre difieren bastante del comportamiento en la rótula, por los motivos que siguen:

- 1- El origen del sistema de coordenadas angulares ya no está alineado con el centro de masas del vehículo, sino que se ubica en la base del dron, pues todo el sistema gira en torno al vértice de la rótula. Esta desalineación provoca que la matriz de inercia del sistema no sea en la práctica diagonal y además aumenta el valor del momento de inercia en los ejes principales.
- 2- Aparece una fuerza más que no se ha considerado en el modelo, el rozamiento con la rótula, cuya magnitud no será despreciable.

Por este motivo, el modelo obtenido se tomará como una mera aproximación y los controladores diseñados empleándolo como referencia habrán de ser posteriormente ajustados sobre el quadrotor. Sin embargo, se trata de un buen punto de partida para el diseño matemático del control.

A continuación, se mostrará una selección de las gráficas comparativas de la variable controlada obtenidas en los ensayos frente a la respuesta que proporciona el modelo obtenido.

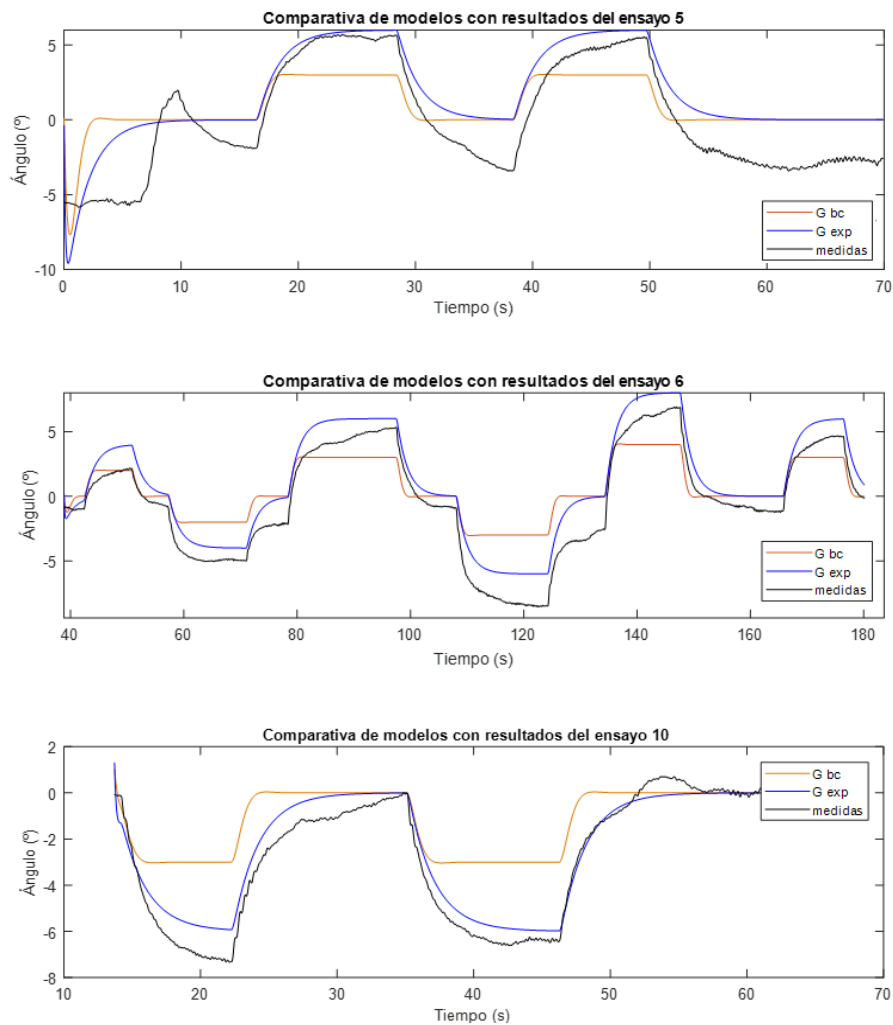


Figura 19 - Gráficas comparativas del modelo con los resultados de los ensayos realizados

5.3. Identificación de la orientación en *yaw*

Para los ensayos del *yaw* emplearemos el mismo controlador del apartado anterior, por ello la estructura de la función de transferencia en bucle cerrado será también:

$$G_{BC} = \frac{1}{\frac{1}{0,009 \cdot K} s^2 + 0,78s + 1} \quad (17)$$

Tras 3 ensayos obtenemos una función de transferencia estimada de:

$$G_{EXP} = \frac{0,9}{0,21s^2 + 0,9s + 1} \quad (19)$$

Comparando los ensayos con los realizados sobre el *pitch* y el *roll* observamos que la *K* del sistema es aproximadamente la misma ($K \simeq 470$), variando esta muy poco entre los experimentos. Además, el término de primer grado del denominador se aproxima mejor a lo esperado y la ganancia está cerca de ser unitaria.

Cabe destacar que, al ser la ganancia estimada más pequeña, que será necesaria una acción de control proporcional más intensa para gobernar este ángulo de giro.

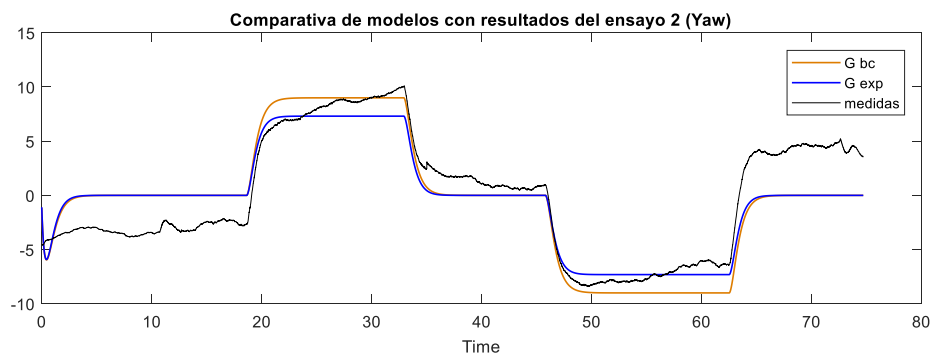


Figura 20 - Gráfica comparativas del modelo con el resultado del ensayo sobre el *yaw*

Sistema de control de la orientación

6.1. Introducción

El objetivo de este apartado es la implementación de un algoritmo de control en bucle cerrado⁴ al subsistema de orientación del quadrotor que logre gobernar su comportamiento de la mejor forma posible.

Esto es algo fundamental, pues del gobierno de este subproceso depende la estabilidad del vehículo. Es especialmente importante el control de los transitorios y el rechazo eficaz de perturbaciones.

Debido a la necesidad de ejecutar el algoritmo en tiempo real a alta frecuencia ($T_s=4ms$) en una Raspberry Pi (ver apartado 3.1.1), se ensayará con técnicas de bajo coste computacional.

Además, se priorizarán aquellas técnicas que posean un ajuste fácil e intuitivo, similar al de los controladores PID.

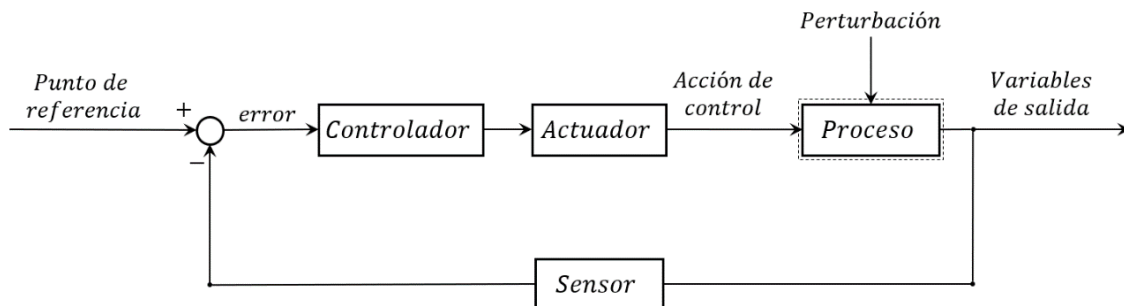


Figura 21 – Esquema general de un sistema de control en bucle cerrado.

6.2. Criterios de diseño

Para el diseño del sistema de control nos guiaremos por los siguientes criterios:

1. Control estable, con las mínimas oscilaciones posibles.
2. Rechazo rápido y eficaz de la perturbación, ya que este aspecto es crítico para mantener el vehículo en vuelo.

⁴ Sistema de control en bucle cerrado: Algoritmo en que la acción de control o variable manipulada se establece en función de la salida medida. Para ello se calcula el error entre el valor de referencia al que se pretende llegar y el valor de la salida, obtenido mediante sensores (realimentación negativa).

3. Error en régimen permanente nulo, pues un error en la orientación provocaría la deriva del sistema e imposibilitaría el control de posición.
4. Control rápido frente a cambios en la referencia (sin que ello comprometa ninguno de los puntos anteriores).

Con esto en mente, probaremos diversas técnicas de control sobre nuestro sistema y realizaremos una evaluación comparativa posterior. El proceso que se ha seguido a la hora de trabajar con cada una ha sido, por seguridad:

1. Cálculo inicial de los parámetros.
2. Implementación del control, expresado como función de transferencia continua, en una simulación y corrección de los parámetros.
3. Implementación del algoritmo de control discreto en una simulación y corrección de los parámetros. Se usará el tiempo de muestreo con el que trabaja nuestro sistema (4ms).
4. Prueba del algoritmo sobre el sistema real en el banco de pruebas.
5. Una vez se haya elegido la técnica de control más adecuada de entre todas las propuestas se realizará una prueba en vuelo libre en el interior del laboratorio.



Figura 22 – Quadrotor anclado a la rótula del banco de pruebas.

Los ensayos realizados en el banco de pruebas tendrán una duración de 50 segundos y consistirán en un escalón en la referencia del proceso seguida de una perturbación en escalón. Dicha perturbación se ha logrado colgando un peso de 70 gramos en uno de los brazos del cuadricóptero.

Se ha dado prioridad al estudio del *roll* y el *pitch* pues su control adecuado es imprescindible para el vuelo del vehículo. Las gráficas adjuntas corresponden a los ensayos realizados sobre el *roll*.

Para llevar esto a cabo, se han creado, en el programa MATLAB®, los scripts, funciones y simulaciones que se adjuntan en el Anexo I – Simulaciones programadas.

6.3. Métodos clásicos

6.3.1. Controlador PID

El algoritmo de control PID es actualmente uno de los controladores más empleados en la industria debido a su relativa simplicidad y al buen conocimiento que se tiene de su comportamiento [26].

Este sistema de control funciona adecuadamente en gran parte de los casos de aplicación industrial, siendo suficiente para alcanzar los requisitos de control deseados.

Su aparición se remonta a la década de 1920 de la mano del matemático Nicolas Minorsky, quien lo desarrolló analizando la dinámica del timón de los buques de guerra estadounidenses.

Su nombre deriva de su estructura, en la que participan las acciones *Proporcional*, *Integral* y *Derivada*:

- La acción proporcional calcula una acción de control sobre el sistema que es proporcional al error de la salida respecto de la referencia.

$$u_p = K_p \cdot e(t) \quad (20)$$

- La acción integral obtiene una acción de control que es proporcional al error acumulado. En esencia, se trata de un control basado en información obtenida de estados anteriores del sistema para el gobierno del estado actual.

$$u_I = K_I \cdot \int_0^T e(t) dt \quad (21)$$

Este control posee una acción más lenta, pero permite obtener un error nulo en régimen permanente y es adecuado para corregir perturbaciones constantes sobre el proceso.

- La acción derivativa actúa cuando se produce un cambio en el valor absoluto del error corrigiéndolo con la misma velocidad con que se produce, evitando de este modo que el error se incremente. Básicamente se trata de un control que se utiliza la tendencia del proceso actual para obtener información del futuro próximo y realizar correcciones rápidas del sistema.

$$u_D = K_D \cdot \frac{de(t)}{dt} \quad (22)$$

Como inconveniente es importante mencionar que la acción derivativa se ve fuertemente afectada por el ruido de medida y por ello es descartada en numerosos sistemas de control.

La estructura de este control en bucle cerrado es la que sigue:

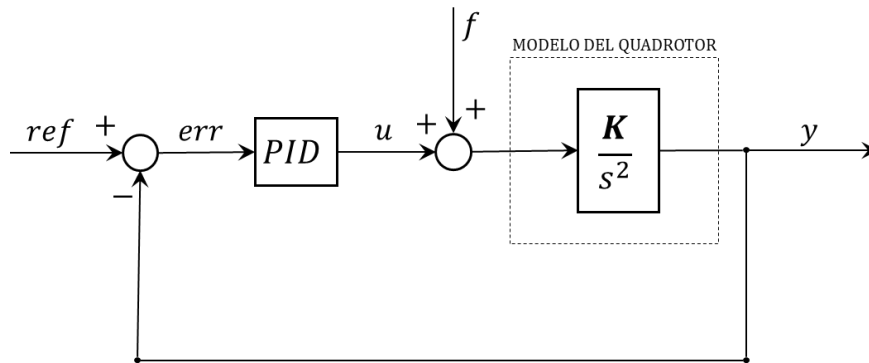


Figura 23 - Estructura del control PID

Siendo la función de transferencia del control:

$$PID = K_c \left[1 + \frac{1}{T_i s} + T_d s \right] = K_p + K_i \frac{1}{s} + K_d s \quad (23)$$

De acuerdo con las ecuaciones (20), (21) y (22):

$$u = u_p + u_i + u_d$$

Siendo su forma discretizada para la implementación en el código:

$$\begin{aligned} u_p(k) &= K_p \cdot (ref(k) - y(k)) = K_p \cdot e(k) \\ u_i(k) &= u_i(k-1) + K_i \cdot e(k) \\ u_d(k) &= K_d \cdot \frac{err(k) - err(k-1)}{T_s} = K_d \cdot de(k) \end{aligned} \quad (23.1)$$

Debido a la naturaleza del proceso (modelado como un doble integrador), la acción derivada se demuestra imprescindible para lograr que el quadrotor se mantenga estable en vuelo.

Trataremos de emplear el controlador más simple posible, en este caso un control *Proporcional - Derivativo* diseñado mediante un análisis heurístico del lugar de las raíces:

Tabla 1 - Parámetros del controlador PD

| Controlador PD: | |
|------------------------|-------|
| K_p | 0,01 |
| K_d | 0.007 |

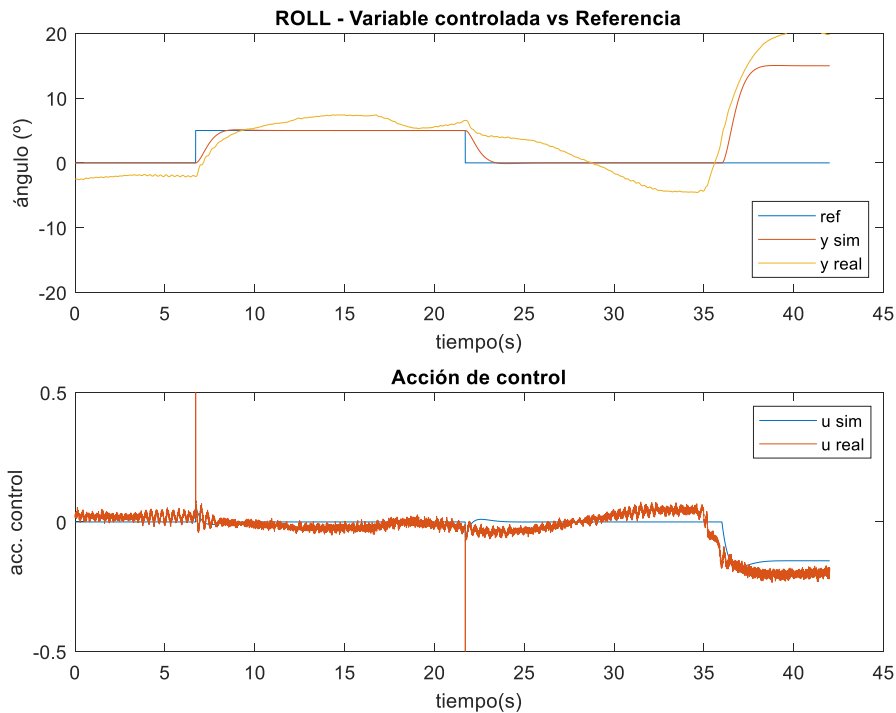


Figura 24 – Gráfica del ensayo del control PD

En la Figura 24 se representan los resultados del experimento para el ángulo *roll*. El experimento es el descrito en la página 45, y se ha realizado de forma paralela en una simulación (*y sim*, *u sim*) y en el banco de pruebas (*y real*, *u real*) de forma que se pueden contrastar los resultados. En la gráfica superior encontramos la referencia angular marcada al sistema comparada con la medida del ángulo *roll*. En la gráfica inferior se muestra la acción de control ejercida. Esta metodología se mantendrá a lo largo de los experimentos realizados en este capítulo.

El control PD estabiliza el sistema consiguiendo un comportamiento aceptable, sin embargo, como cabía esperar, no es capaz de realizar un buen seguimiento de la referencia pues la acción proporcional no basta para eliminar el error en régimen permanente. Además, este control tampoco es capaz de compensar la perturbación en escalón introducida.

Para solucionar esto introduciremos acción integral al algoritmo. No se ha podido dar mucho peso a esta componente debido a las oscilaciones que producía en el proceso en bucle cerrado. Además, ha sido necesario realizar un pequeño reajuste al resto de parámetros para minimizar esto.

Tabla 2 – Parámetros del controlador PID

| Controlador PID: | |
|-------------------------|--------------------|
| K_p | 0,017 |
| K_d | 0.006 |
| K_i | 3·10 ⁻⁵ |

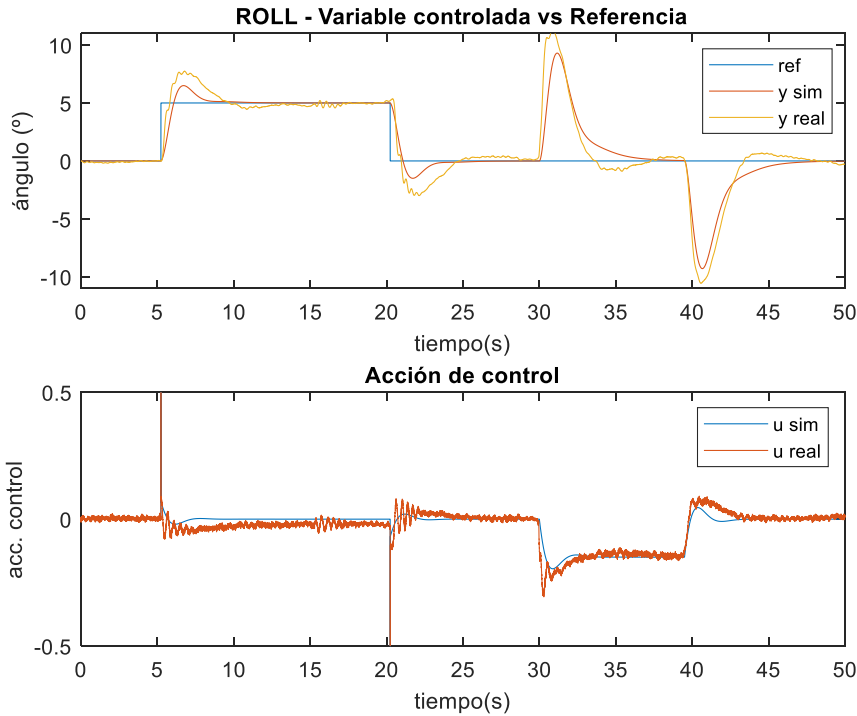


Figura 25 – Gráfica del ensayo del control PID

Con la acción integral el sistema logra ajustarse a la referencia durante la mayoría del tiempo y compensar la perturbación. Sin embargo, el transitorio generado por la perturbación es demasiado largo y aparece una sobreoscilación apreciable en el seguimiento de la referencia.

Respecto a este segundo problema, disponemos de dos métodos sencillos de solucionarlo. Procederemos a aplicarlos a continuación.

6.3.2. Controlador PID con filtro en la referencia

Para suprimir la sobreoscilación generada por el PID añadiremos un filtro sobre la referencia buscando suavizar la pendiente de las variaciones sobre la misma y reducir de este modo los picos que experimenta la acción de control.

$$ref_{FILTRADA} = \frac{\overbrace{1}^{\text{filtro}}}{\alpha s + 1} \cdot ref$$

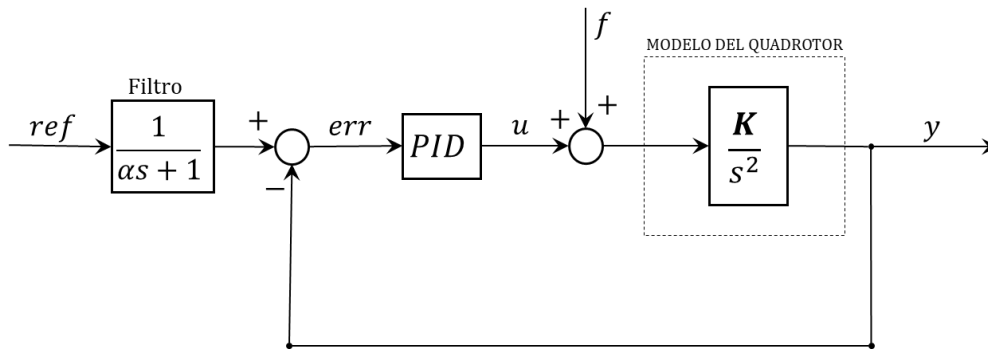


Figura 26 - Estructura del control PID con filtrado de la referencia

Donde a α se le ha dado el valor 0,8 elegido de forma empírica.

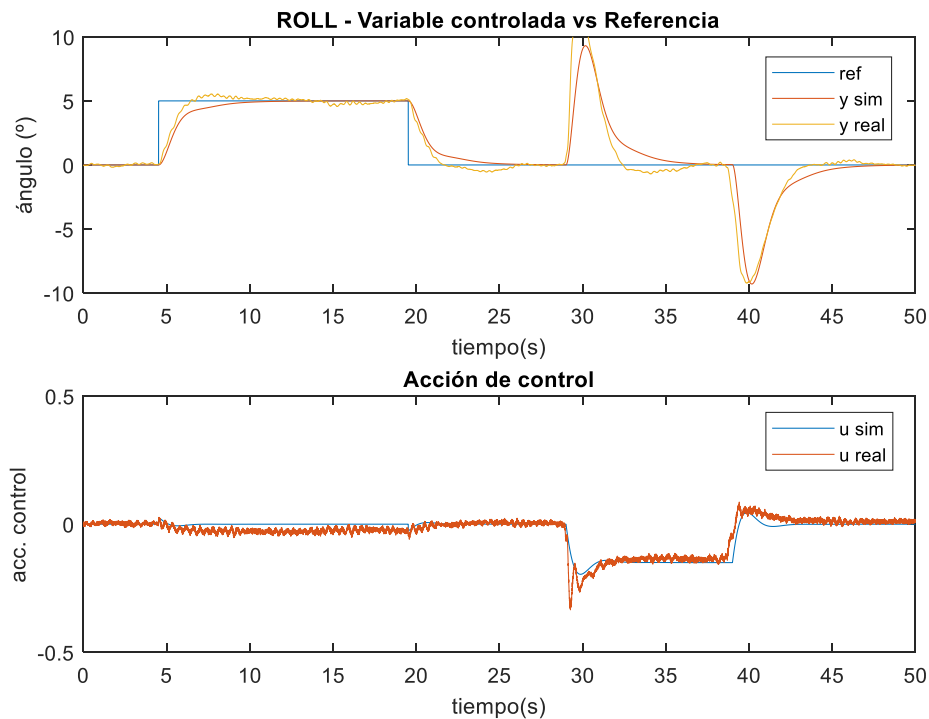


Figura 27 - Gráfica del ensayo del control PID con filtrado en la referencia

6.3.3. Controlador PID con dos grados de libertad

Recurriremos a este algoritmo pues se trata de otro modo de solucionar las sobreoscilaciones producidas por el controlador PID.

Esta estructura busca desacoplar la respuesta frente a cambios en la referencia de la respuesta frente a perturbaciones externas mediante ponderación de la referencia. Esto, además, tiene la ventaja de que los parámetros del control son mucho más sencillos de ajustar.

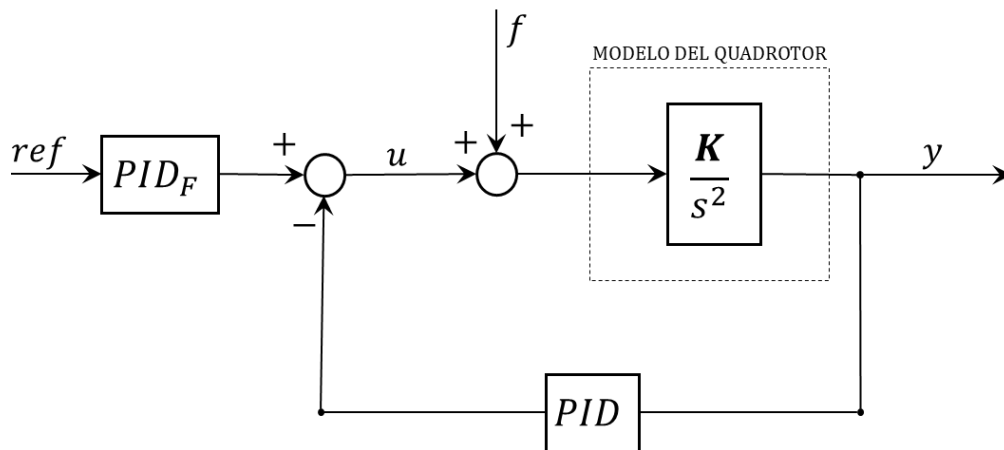


Figura 28 - Estructura del control PID de dos grados de libertad

$$\text{En la que } PID_F = K_c \left[\mathbf{b} + \frac{1}{T_i s} + \mathbf{c} T_d s \right] \quad (24)$$

Lo que, discretizando, resulta en:

$$\begin{aligned} u_p(k) &= K_p \cdot (\mathbf{b} \cdot \text{ref}(k) - y(k)) \\ u_I(k) &= u_I(k-1) + K_I \cdot \text{err}(k) \\ u_D(k) &= K_D \cdot (\mathbf{c} \cdot \text{dref}(k) - dy(k)) \\ u &= u_p + u_I + u_D \end{aligned} \quad (24.1)$$

Siendo b y c los parámetros que emplearemos para ponderar la referencia. Existen diversos métodos para lograr un mejor comportamiento del sistema modificando la función de transferencia en bucle cerrado, pero nosotros optaremos por la vía más sencilla dándoles valor cero.

Tabla 3 – Controlador PID de dos grados de libertad

Controlador PID (2dof):

| | |
|----------------------|--------------------|
| K_p | 0,017 |
| K_d | 0.006 |
| K_i | 3·10 ⁻⁵ |
| b, c | 0 |

Al anular los parámetros b y c nuestro objetivo es eliminar los ceros del numerador en la función de transferencia que relaciona la salida con la referencia introducida y con ello las sobreoscilaciones. Con ello obtendremos resultados similares a los que ofrecía el filtrado en la referencia del PID.

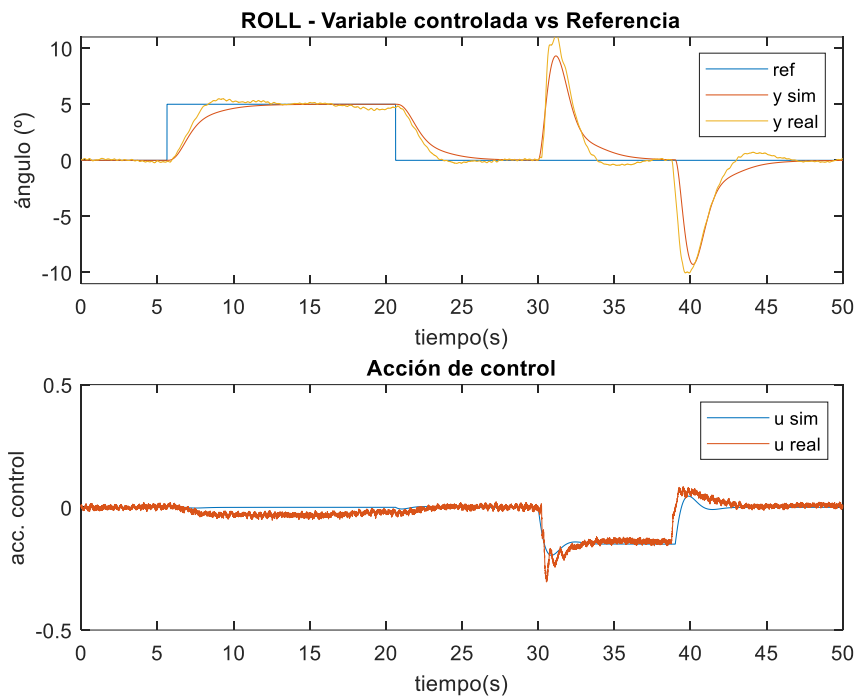


Figura 29 – Gráfica del ensayo del control PID con dos grados de libertad

En los algoritmos de control anteriores, pese a haber solucionado las sobreoscilaciones de la respuesta frente a cambios en la referencia, la compensación de las perturbaciones sigue realizándose de forma muy lenta en comparación con la dinámica real del quadrotor.

La acción integral, que al darle más peso podría mejorar esto, tiene valores ínfimos pues aumentarla compromete la estabilidad del sistema ya que lo vuelve muy oscilatorio.

Exploraremos por tanto otras opciones que no tengan dichos inconvenientes y que puedan sustituir a la componente integral.

6.4. Estimación del estado

6.4.1. Observadores:

El observador de Luenberger es un algoritmo que permite aproximar el estado interno no medible de un sistema dinámico lineal a partir de sus entradas y salidas y de su estado en el pasado [27].

El principio básico del observador es el uso de un sistema virtual que trata de ser una copia del proceso real a controlar. De este sistema ficticio sí conocemos perfectamente su estado interno y buscamos que, sometido a los mismos estímulos que el sistema real, su comportamiento converja al de este. Para lograrlo, el sistema virtual recibirá la entrada del sistema real corregida con el error existente entre la salida del sistema virtual y la salida del sistema real multiplicado por una constante (L).

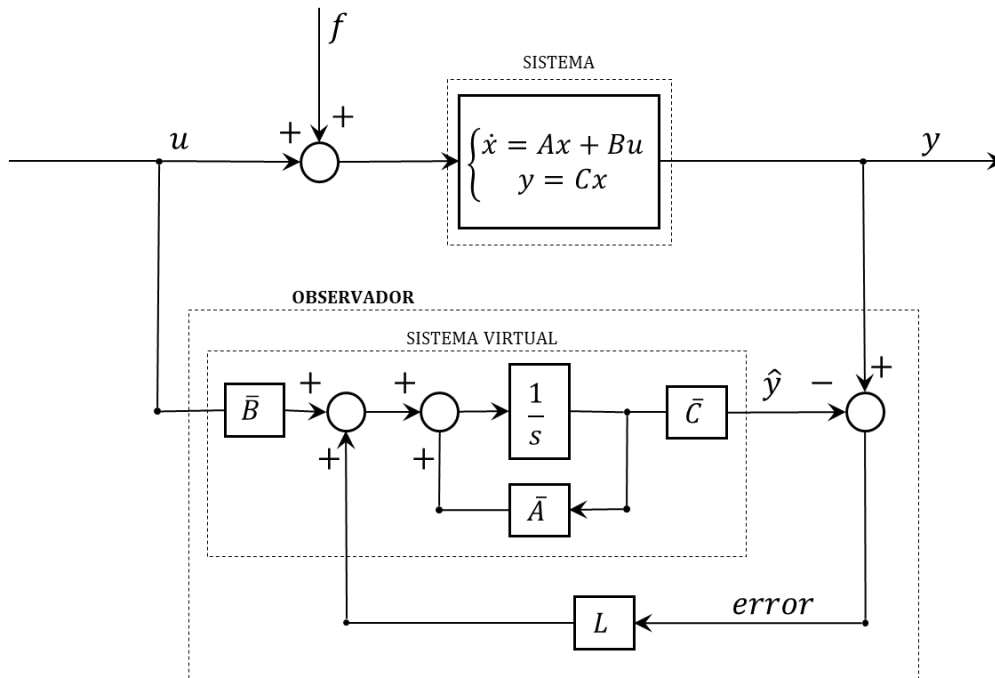


Figura 30 - Estructura de un observador

Según nuestro modelo (16), el estado del proceso posee la forma:

$$\dot{x} = Ax + Bu$$

De acuerdo con el observador (ver Figura 30), el estado estimado será:

$$\hat{\dot{x}} = \bar{A}\hat{x} + \bar{B}u + L \underbrace{(y - \bar{C}\hat{x})}_{error}$$

↓

$$\hat{\dot{x}} = (\bar{A} - L\bar{C})\hat{x} + \bar{B}u + Ly \quad (25)$$

Sabiendo que \bar{A} , \bar{B} y \bar{C} son parámetros del modelo teórico en espacio de estados.

Idealmente aspiraremos a que el error entre el estado aproximado y el estado real tienda a ser nulo.

$$\begin{aligned} \dot{e} &= \dot{x} - \dot{\hat{x}} \\ \dot{e} &= Ax + Bu - \bar{A}\hat{x} - \bar{B}u - L(y - \bar{C}\hat{x}) \\ \dot{e} &= Ax - \bar{A}\hat{x} - L(Cx - \bar{C}\hat{x}) \\ &\quad \downarrow \bar{A} \simeq A, \bar{C} \simeq C \\ \dot{e} &= (\bar{A} - L\bar{C})e \end{aligned} \tag{26}$$

Al ser este un sistema dinámico sin entradas externas, siempre y cuando la matriz $(\bar{A} - L\bar{C})$ sea estable, la única solución posible a (26) será la trivial (el error tenderá a ser nulo). Para ello será imprescindible diseñar L tal que se cumpla la condición mencionada.

6.4.2. ESO (Extended State Observer) – Predicción de la perturbación

Al estar trabajando con un vehículo aéreo, es de gran importancia tener en cuenta que este estará afectado por perturbaciones que podrían desestabilizar el proceso:

$$\dot{x} = Ax + Bu + Bf \tag{27}$$

Disponer de una medida de estas sería muy conveniente ya que nos permitiría prealimentarla en el sistema y de ese modo compensar su efecto.

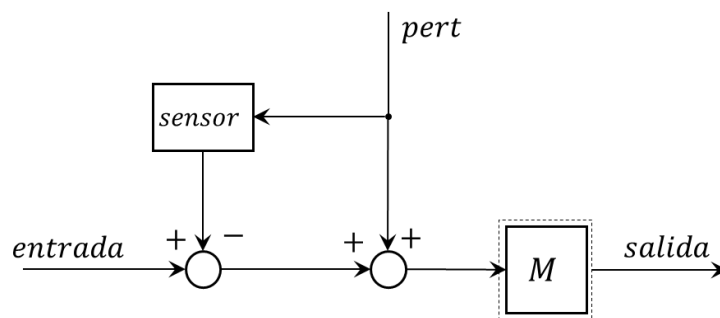


Figura 31 – Esquema de la prealimentación de la perturbación

Dado que no se dispone de tecnología para realizar esto, se tratará de predecir el valor de las perturbaciones considerándolas como parte del estado del sistema y empleando un observador.

Para lograr esto se empleará el *Observador de Estados Extendido (E.S.O.)*, elemento principal del *Active Disturbance Rejection Control (ADRC)* un nuevo paradigma de control que tiene potencial para desplazar al popular PID y hacerse un hueco en el mercado.

El ESO se basa en el observador de Luemberger, pero extiende el tamaño del estado interno del sistema incluyendo en él la perturbación estimada (\hat{f}) [28].

$$\dot{z} = \begin{bmatrix} \dot{x} \\ \dot{\hat{f}} \end{bmatrix} = \underbrace{\begin{bmatrix} A & B \\ 0 & 0 \end{bmatrix}}_{\mathbb{A}} \begin{bmatrix} x \\ \hat{f} \end{bmatrix} + \underbrace{\begin{bmatrix} B \\ 0 \end{bmatrix}}_{\mathbb{B}} u + \begin{bmatrix} 0 \\ 1 \end{bmatrix} \dot{f}$$

Por simplicidad supondremos que el sistema se ve afectado por una perturbación constante o en escalón. Esto nos permitirá considerar que $\dot{f} \simeq 0$.

$$\dot{z} = \mathbb{A}z + \mathbb{B}u + \emptyset$$

El estado estimado tendrá entonces la forma:

$$\begin{aligned} \dot{\hat{z}} &= \bar{\mathbb{A}}\hat{z} + \bar{\mathbb{B}}u + L \underbrace{(y - \bar{\mathbb{C}}\hat{z})}_{\text{error}} \\ \hat{z} &= (\bar{\mathbb{A}} - L\bar{\mathbb{C}})\hat{z} + \bar{\mathbb{B}}u + Ly \end{aligned} \quad (28)$$

↓

$$\begin{bmatrix} \hat{x}_1 \\ \hat{x}_2 \\ \hat{f} \end{bmatrix} = \begin{bmatrix} -L_1 & 1 & 0 \\ -L_2 & 0 & K \\ -L_3 & 0 & 0 \end{bmatrix} \begin{bmatrix} \hat{x}_1 \\ \hat{x}_2 \\ \hat{f} \end{bmatrix} + \begin{bmatrix} 0 \\ K \\ 0 \end{bmatrix} u + \begin{bmatrix} L_1 \\ L_2 \\ L_3 \end{bmatrix} y \quad (28.1)$$

De la solución de esta ecuación matricial obtendremos el valor de $\hat{f} \equiv \hat{z}_3$, que podremos prealimentar al sistema para compensar la perturbación.

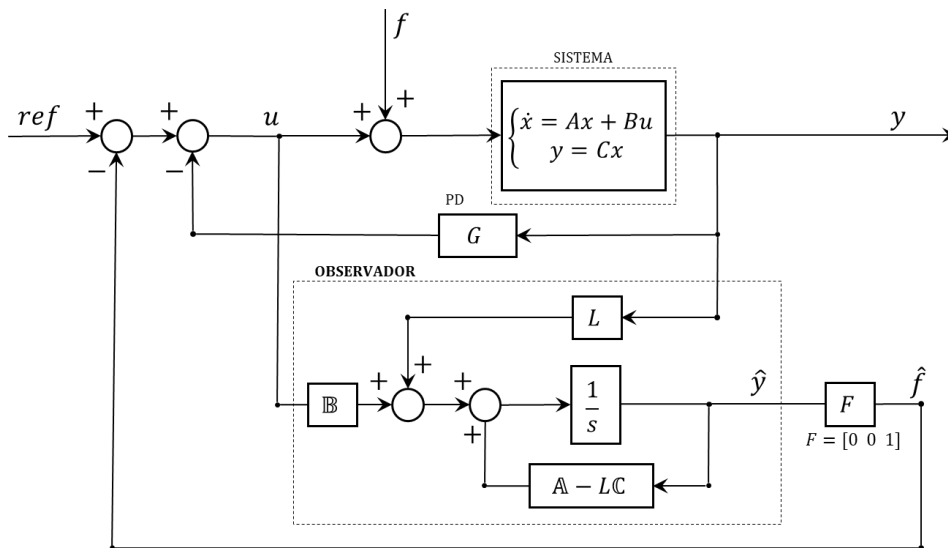


Figura 32 - Estructura del control PD + ESO

Discretización de la ecuación (28) para la programación del algoritmo:

$$\hat{z}_1(k) = (1 - L_1 \cdot Ts) \cdot \hat{z}_1(k-1) + Ts \cdot \hat{z}_2(k-1) + L_1 \cdot Ts \cdot y(k) \quad (28.2)$$

$$\hat{z}_2(k) = -L_2 \cdot Ts \cdot \hat{z}_1(k-1) + \hat{z}_2(k-1) + K \cdot Ts \cdot \hat{z}_3(k-1) + K \cdot Ts \cdot u(k-1) + L_2 \cdot Ts \cdot y(k)$$

$$\hat{z}_3(k) = -L_3 \cdot Ts \cdot \hat{z}_1(k-1) + \hat{z}_3(k-1) + L_3 \cdot Ts \cdot y(k)$$

$$\hat{f}(k) = \hat{z}_3(k)$$

El observador habrá de acompañarse con un control PD (o por un control por realimentación del estado equivalente) para lograr mantener el sistema en equilibrio.

El ajuste del ESO como algoritmo de control es un compromiso entre el rechazo de la perturbación y la amplificación del ruido de medida, con la ventaja de que está totalmente desacoplado del control PD y no compromete la estabilidad del sistema.

Este ajuste se realiza, como se ha explicado en el apartado anterior, a través del vector L , ubicando los polos (autovalores) de la matriz $(\bar{A} - L\bar{C})$ (26). Se trata de un problema de optimización: a medida que hagamos el control más rápido la estimación de la perturbación será mejor, pero la acción de control se verá más afectada por el ruido de la medida (Ver Figura 33 y Figura 34).

En la siguiente simulación, sin los efectos del ruido, se aprecia el potencial de este algoritmo:

Polos en -1, -2 y -3 (control lento):

Polos en -50, -52, -53 (control rápido):

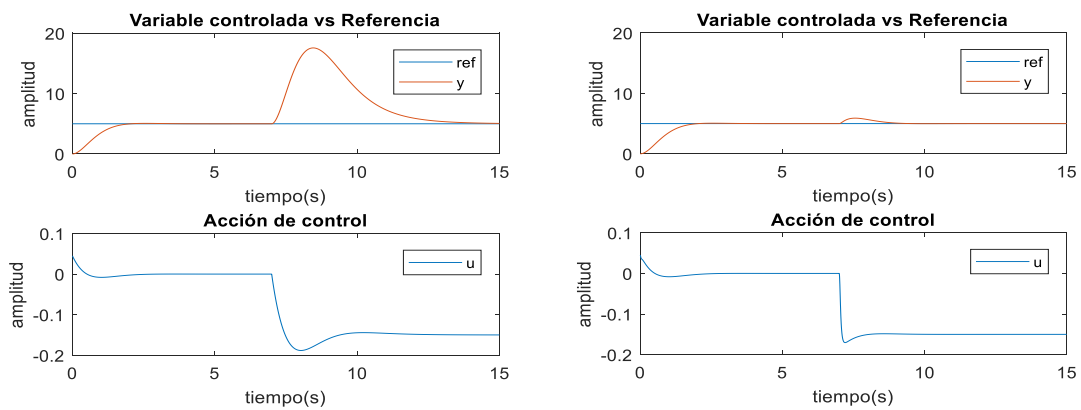


Figura 33 - Simulación del potencial del ESO (sin ruido en la medida)

Por otro lado, en esta otra simulación se ha añadido ruido blanco al bucle de realimentación de la derivada:

Control lento:

Control rápido:

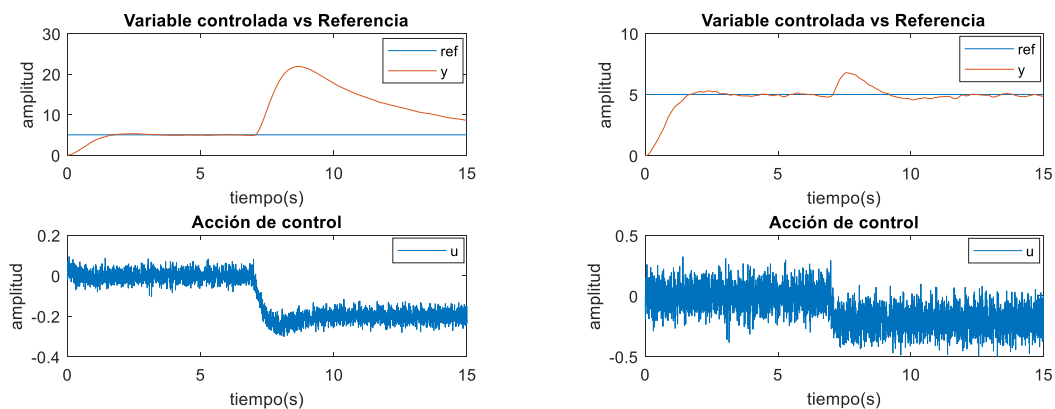


Figura 34 - Efecto del ruido blanco sobre el control ESO y UDE

Adicionalmente se agregará un filtro sobre la referencia ($\alpha=0,8$) para evitar la sobreoscilación, aunque ralentizará levemente el control del sistema.

Tabla 4 – Parámetros del controlador PD + ESO

| Controlador PD + ESO: | |
|------------------------------|-------------------------------------|
| K_p | 0,01 |
| K_d | 0.007 |
| L | [15 74 0,257] |
| α | 0,8 $\xrightarrow{T_s=0,004}$ 0,995 |

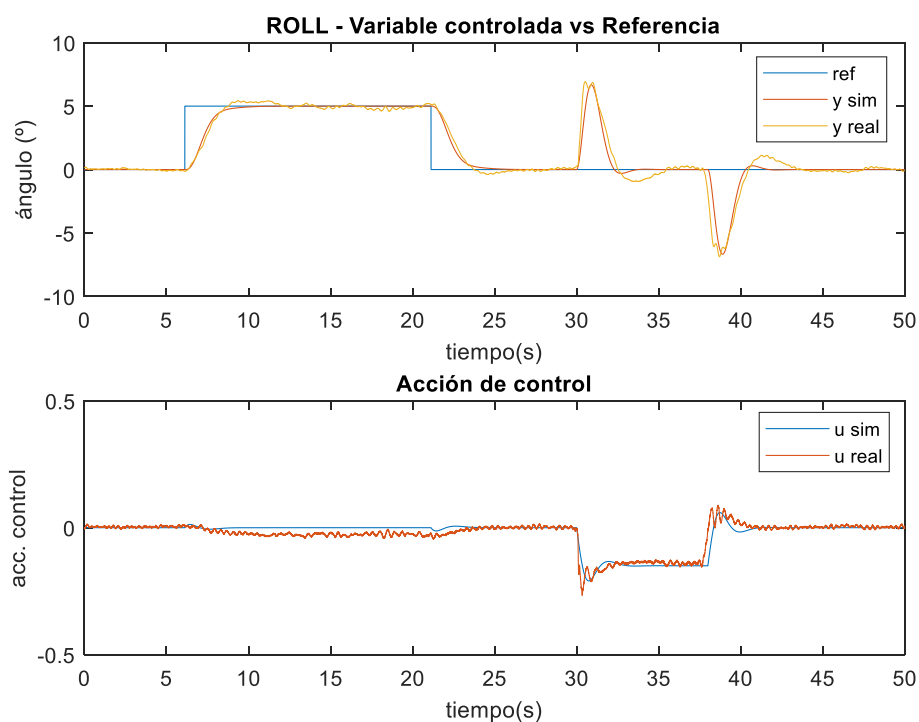


Figura 35 – Gráfica del ensayo del control PD + ESO

De la gráfica se desprende que la respuesta frente a las perturbaciones es mucho más efectiva y los picos máximos que esta produce sobre la variable controlada se reducen apreciablemente.

Como desventaja, es preciso comentar que el ajuste de este algoritmo es complejo pues se ha de modificar L_1 , L_2 y L_3 simultáneamente para ubicar los polos de la matriz en el lugar deseado. Se demuestra preciso trabajar sobre un modelo aproximado del sistema previamente a los ensayos reales.

6.5. Realimentación del estado

Poseer medidas del estado completo del sistema en cada momento nos permite aplicar un control mucho más preciso pues podremos emplear la medida de la velocidad angular del sistema en sustitución de la aproximación lineal de la derivada.

Dado que la IMU obtiene medidas de la orientación y de la velocidad angular del quadrotor en cada en cada ciclo nos es posible aplicar técnicas de control que las empleen.

6.5.1. PID de dos grados de libertad con medición de la derivada

Buscando comprobar los beneficios de la realimentación del estado sobre el controlador PID sustituiremos la aproximación lineal de la derivada que este utiliza por la medición obtenida.

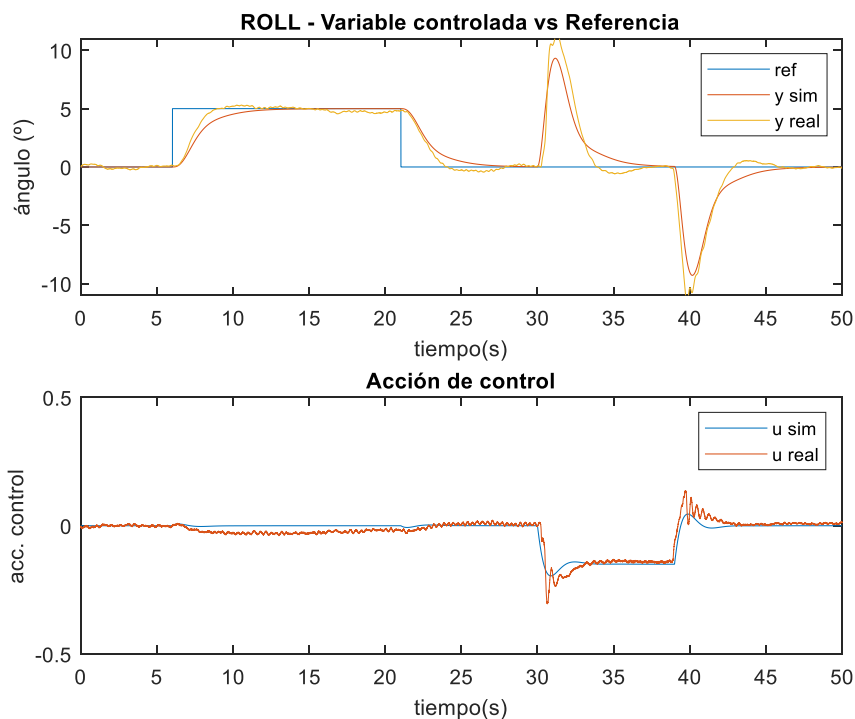


Figura 36 – Gráfica del ensayo del control PID de 2 g.d.l. con medición de la derivada

Pese a que la respuesta frente a perturbaciones sigue siendo lenta se aprecia un menor impacto del ruido de medida en la acción de control, siendo las oscilaciones de la misma mucho más reducidas.

6.5.2. UDE

Rigiéndose el modelo del sistema a controlar por la ecuación (27):

$$\dot{x} = Ax + Bu + Bf \quad (27)$$

Al disponer de medidas del estado completo nuestra única incógnita es el valor de la perturbación (f) y, por tanto, nos será posible despejarlo y prealimentarlo. Al hacer esto consideraremos también cualquier divergencia del proceso real con el modelo matemático como una perturbación y el control tratará de compensarla.

Se trata por tanto de un algoritmo de gran robustez frente a errores de modelado y al no suponer la perturbación como constante poseerá una dinámica más veloz. A esta técnica se la conoce como *Estimador de Perturbación e Incertidumbre (UDE)* [29].

$$f \simeq \hat{f} = B^\dagger[\dot{x} - Ax - Bu]$$

↓ Transformada de Laplace

$$\hat{f} = B^\dagger[sx - Ax - Bu] \quad (29)$$

La expresión obtenida no es implementable y será necesario añadirle un filtro.

$$\hat{f} = \frac{1}{\tau s + 1} B^\dagger[sx - Ax - Bu] \quad (29.1)$$

Particularizando dicha expresión al modelo del quadrotor obtenemos:

$$\hat{f} = \frac{s \cdot K^{-1}}{\tau s + 1} y - \frac{1}{\tau s + 1} u \quad \text{donde } K = B_2 \quad (30)$$

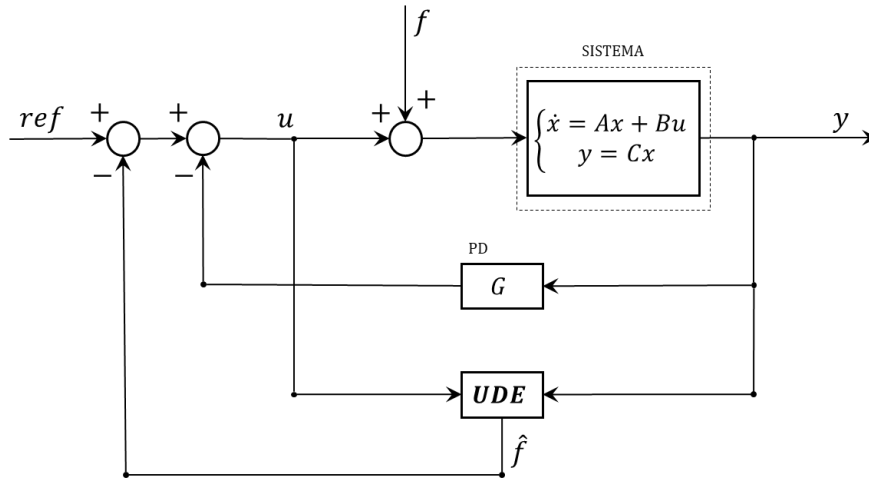


Figura 37 - Estructura del control PD + UDE

Discretizando la expresión anterior:

$$\hat{f}_u(k) = \tau \cdot \hat{f}_u(k-1) - (1-\tau) \cdot u(k-1) \quad (30.1)$$

$$\hat{f}_y(k) = \tau \cdot \hat{f}_y(k-1) + \beta \cdot dy(k) - \beta \cdot dy(k-1) \quad \text{siendo } \beta = (1-\tau)/(Ts \cdot K)$$

$$\hat{f}(k) = \hat{f}_u(k) + \hat{f}_y(k)$$

Este algoritmo habrá de acompañarse también de un control por realimentación del estado (equivalente a un PD) para controlar el quadrotor.

El ajuste del UDE, al igual que el del ESO, es un compromiso entre el rechazo de la perturbación y la amplificación del ruido de medida, estando también totalmente desacoplado del control PD.

Este ajuste se realizará a través del filtro aplicado (optimización del parámetro τ).

El mejor resultado que hemos podido obtener sobre el quadrotor atendiendo a este compromiso es una τ de 0,8. Se emplea también un filtro sobre la referencia ($\alpha=0,8$) para evitar la sobreoscilación.

Tabla 5 - Parámetros del controlador PD + UDE

| Controlador PD + UDE: | |
|------------------------------|-------------------------------------|
| K_p | 0,01 |
| K_d | 0.007 |
| τ | 0,8 $\xrightarrow{T_s=0,004}$ 0,995 |
| α | 0,8 $\xrightarrow{T_s=0,004}$ 0,995 |

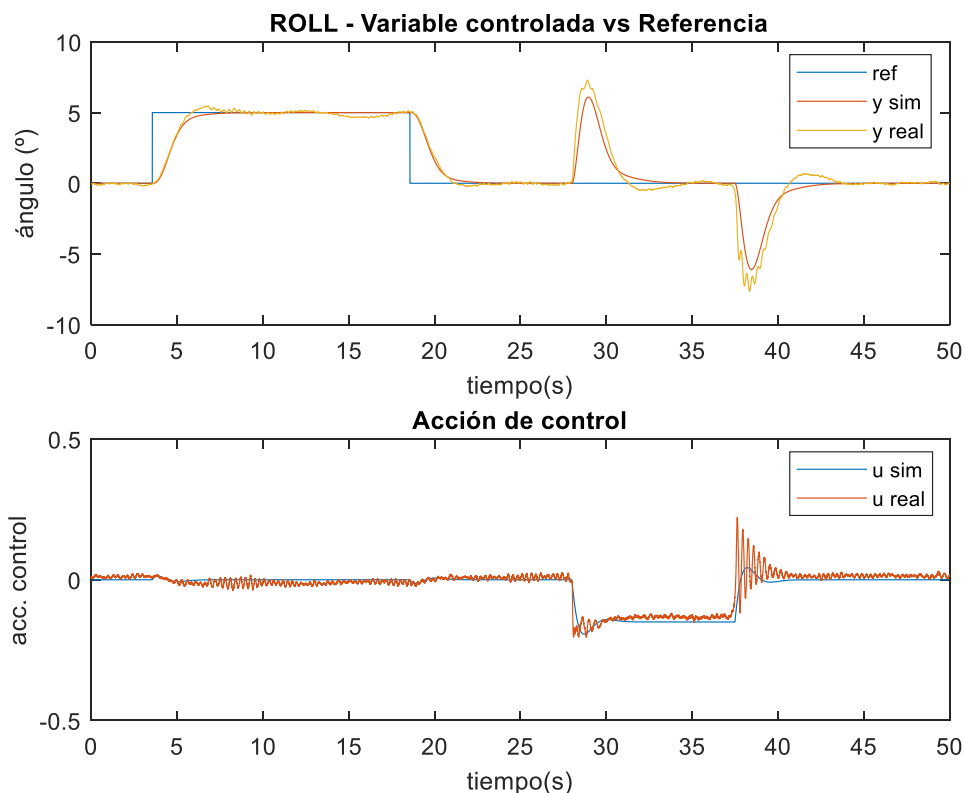


Figura 38 – Gráfica del ensayo del control PD + UDE

La respuesta obtenida es muy similar a la del ESO, logrando una reducción apreciable del impacto de las perturbaciones. Además, observamos que se reduce levemente la sobreoscilación de la variable controlada al compensar la perturbación, pero a su vez aumentan las oscilaciones de la acción de control.

Cabe destacar además que, al solo tener que ubicar el polo del filtro, el ajuste del UDE es mucho más sencillo que el de los anteriores algoritmos pudiendo realizarse este *in situ*.

Nótese en las gráficas mostradas las no-linealidades del sistema que controlamos. Estas se pueden ver, por ejemplo, en la diferencia existente entre las respuestas del proceso frente a una perturbación positiva y otra negativa. Pese a esto, los controladores han respondido bien demostrando su robustez.

6.6. Conclusiones del apartado:

6.6.1. Elección del algoritmo de control

A pesar de que el sistema estudiado posee una dinámica rápida e inestable, el tradicional control PID se ha manifestado como una técnica válida para su control. Sin embargo, la componente integral del algoritmo posee dos características poco ventajosas para un proceso de esta índole. Estas son:

- La presencia de un polo en el origen, resultado del integrador (1/s), que posee una dinámica relativamente lenta y oscilatoria, capaz de desestabilizar el vehículo.
- Al introducir acción integral se hace necesario recalibrar el resto de los parámetros. Su efecto sobre el sistema no es independiente de las acciones de control complementarias (proporcional y derivada) aplicadas sobre el mismo, por tanto, el ajuste del algoritmo requiere nociones teóricas previas.

Por estos motivos hemos sustituido la acción integral por técnicas de estimación y realimentación del estado. Estos algoritmos solo se ven limitados por el ruido de medida y logran regímenes transitorios superiores. Estas técnicas únicamente podrían desestabilizar el sistema si se eligen parámetros que resultan en un control demasiado rápido, con mucha amplificación del ruido. Pero la solución es sencillamente elegir parámetros para un control más lento, ya que la acción de estos algoritmos está desacoplada del comportamiento dinámico del sistema.

Las técnicas UDE y ESO muestran un desempeño similar en los ensayos realizados, reduciendo la amplitud de las perturbaciones y mejorando la estabilidad del vehículo. Sin embargo, el UDE posee la clara ventaja de su simplicidad a la hora de ajustar y programar el algoritmo.

Por tanto, para su empleo en la siguiente fase, el ensayo en vuelo libre, nos decantaremos por la estructura de control PD+UDE pues, pese a la presencia de cierto ruido en la acción de control, este posee una amplitud asumible y el ajuste del algoritmo es muy sencillo, ya que se encuentra totalmente desacoplado de la dinámica del PD y solo se ha de modificar un parámetro (el polo del filtro).

Para el control de los ángulos *pitch* y *roll* se emplearán los parámetros especificados en la *Tabla 5 - Parámetros del controlador PD + UDE*. Por último, para el ángulo *yaw* se ha realizado un ajuste empírico partiendo de dichos parámetros y se ha implementado el siguiente control PD+UDE:

Tabla 6 – Parámetros del controlador PD+UDE del yaw

| Controlador PD + UDE | |
|-----------------------------|-------------------------------------|
| K_p | 0,02 |
| K_d | 0.007 |
| τ | 0,8 $\xrightarrow{T_s=0,004}$ 0,995 |
| α | 0,8 $\xrightarrow{T_s=0,004}$ 0,995 |

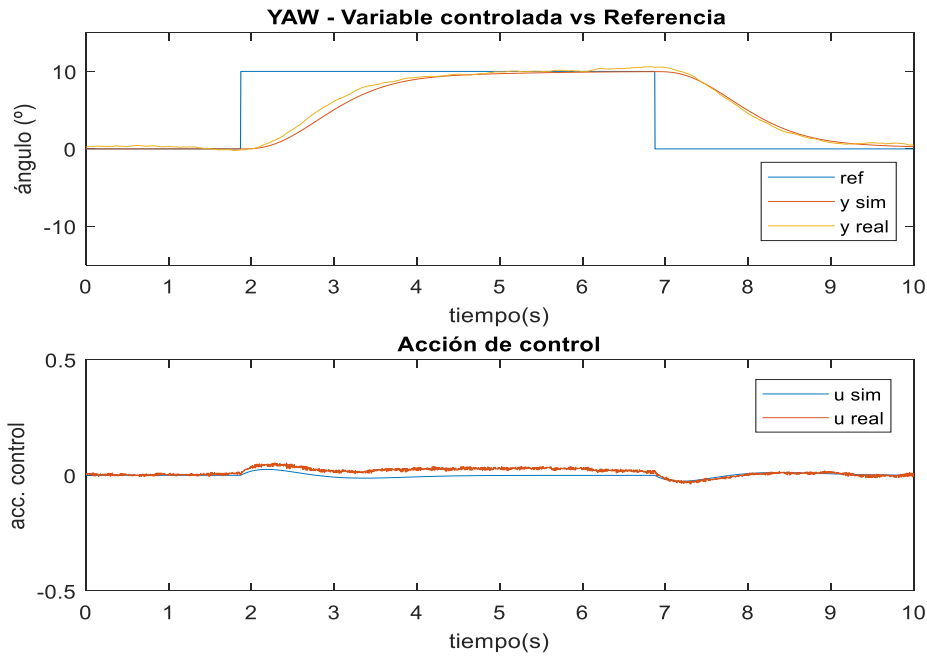


Figura 39 – Gráfica del ensayo del control PD+UDE del yaw

6.6.2. Adaptación del diseño al vuelo libre

Para que los algoritmos de control diseñados sobre el banco de trabajo funcionen adecuadamente en vuelo *hover* se habrá de escalar la constante proporcional del PD de forma empírica para lograr un control algo menos agresivo.

Esto es así debido a las diferencias de comportamiento entre el dron anclado a la rótula y el dron en vuelo. Como se ha explicado en el apartado 5.2, la inercia del sistema en el banco de pruebas es mucho mayor que en vuelo libre y, según las ecuaciones de nuestro modelo simplificado (14), (15), la inercia del vehículo se relaciona directamente con la constante (K) del modelo del sistema de orientación.

$$\ddot{\varphi} \simeq \frac{1}{I_{xx}} \tau_{\varphi} + f(\dot{\varphi}, \dot{\theta}, \dot{\psi}, d_{externa}) \longrightarrow s^2 \theta \simeq \mathbf{K} u + f$$

Por esta razón, al reducirse la inercia del sistema, la constante proporcional del control habrá de reducirse de igual modo para no inducir oscilaciones o incluso inestabilidad al cuadricóptero.

Se multiplicará entonces la constante proporcional del PD por un coeficiente de 0,4 en el control de los tres ángulos. Dicho coeficiente se ha obtenido observando el comportamiento previo al despegue del dron.

Sistema de control de la posición

7.1. Contextualización

El siguiente paso lógico tras haber logrado la estabilidad del dron es el diseño de un control en cascada sobre el anterior que gobierne la posición del vehículo en las tres dimensiones del espacio.

De acuerdo con el modelo de Euler-Lagrange para el subsistema traslacional (ver apartado Ecuación del modelo traslacional: del Capítulo 4) las ecuaciones que gobiernan el desplazamiento en los ejes x e y (12) pueden aproximarse también a la estructura de doble integrador empleada en el modelo de orientación:

$$s^2 \cdot pos \simeq K \theta + f \quad \text{siendo } K \simeq \frac{1}{\text{masa}}$$

Como la masa del quadrotor es de aproximadamente 1kg, supondremos una $K \simeq 1$ para el diseño del control.

Para el sensado de la posición del cuerpo en el interior del laboratorio se utilizará el sistema de visión artificial *Optitrack* explicado en el apartado 3.5, que nos proporciona una medición fiable con un error del orden de milímetros.

Se ha configurado el periodo de envío de los datos de medición de la posición a 100ms pues, de acuerdo con nuestro modelo, la dinámica de este subproceso es mucho más lenta que la del subsistema de orientación y no es necesaria mayor frecuencia para la ejecución del algoritmo.

A diferencia de los ensayos del apartado anterior, se empleará una estructura de control *en equis*, debido a los beneficios que esta aporta (ver apartado 4.4). Además, se fijará la referencia del ángulo *yaw* a cero para lograr que los planos coordenados $\bar{X}\bar{Z}$ e $\bar{Y}\bar{Z}$ del sistema de referencia relativo al cuerpo coincidan, en la medida de lo posible, con sus homólogos del sistema de coordenadas absolutas. Esto simplificará el control del proceso evitando la necesidad de usar matrices de transformación para rotar las medidas del *Optitrack* (ver Figura 44 – Gráfica del control del yaw).

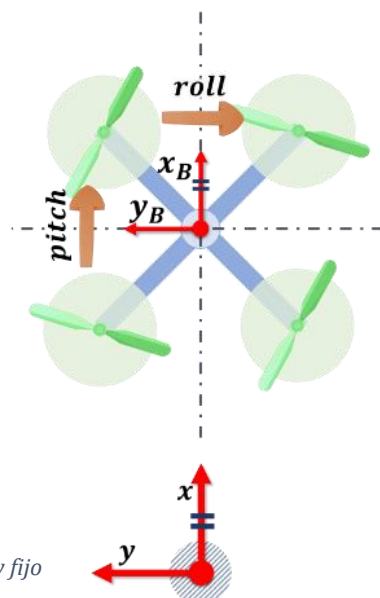


Figura 40 - Estructura del quadrotor en equis con el yaw fijo

Para el ajuste y la validación del control de posición sobre el vehículo no disponemos de ningún sistema auxiliar que nos permita realizarlo de forma segura, habiendo de realizar este proceso con el dron en vuelo. Por este motivo no se ha podido realizar el proceso de identificación pertinente. Es imprescindible por tanto que el método utilizado tenga un ajuste lo más rápido y sencillo posible.

7.2. Algoritmo empleado

Viendo los buenos resultados obtenidos con el algoritmo UDE en el apartado anterior frente a un sistema también modelado como un doble integrador, y sumando a esto su fácil ajuste *in situ* y su robustez frente a errores de modelado y perturbaciones, nos decantaremos por esta técnica.

El esquema del control en cascada se muestra a continuación:

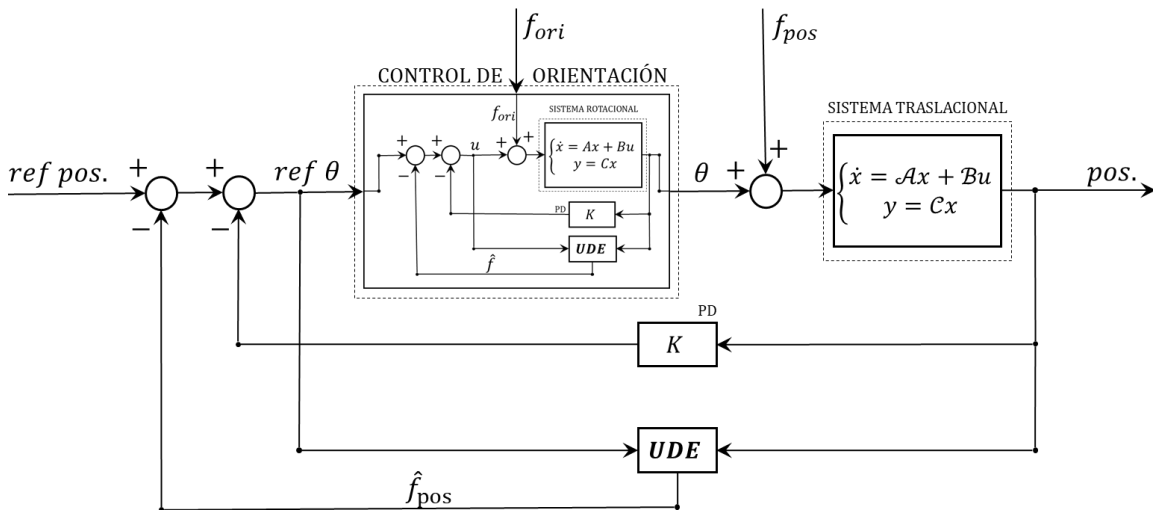


Figura 41 – Esquema del control de posición en los ejes X e Y (PD+UDE)

El proceso de diseño ha sido el siguiente: el diseño inicial del controlador PD se ha realizado usando la herramienta *rltool* (*Control System Designer*) de MATLAB® [30], y se ha probado sobre un modelo diseñado en *Simulink*⁵ antes de reajustarlo sobre el quadrotor. Finalmente, sobre el control PD ya estable en el dron se ha ajustado la acción del UDE.

Los resultados se aprecian en las figuras de la página siguiente (Figura 42 y Figura 43).

Tabla 7 – Parámetros del control PD+UDE en los ejes X e Y

| Controlador PD + UDE | |
|-----------------------------|----------------------------------|
| K_p | 3 |
| K_d | 7 |
| τ | 3,3 $\xrightarrow{T_s=0,1}$ 0,97 |

⁵ En el Anexo I – Simulaciones programadas se incluye una breve explicación del modelo diseñado en MATLAB®.

Control del *eje x* (PD+UDE):

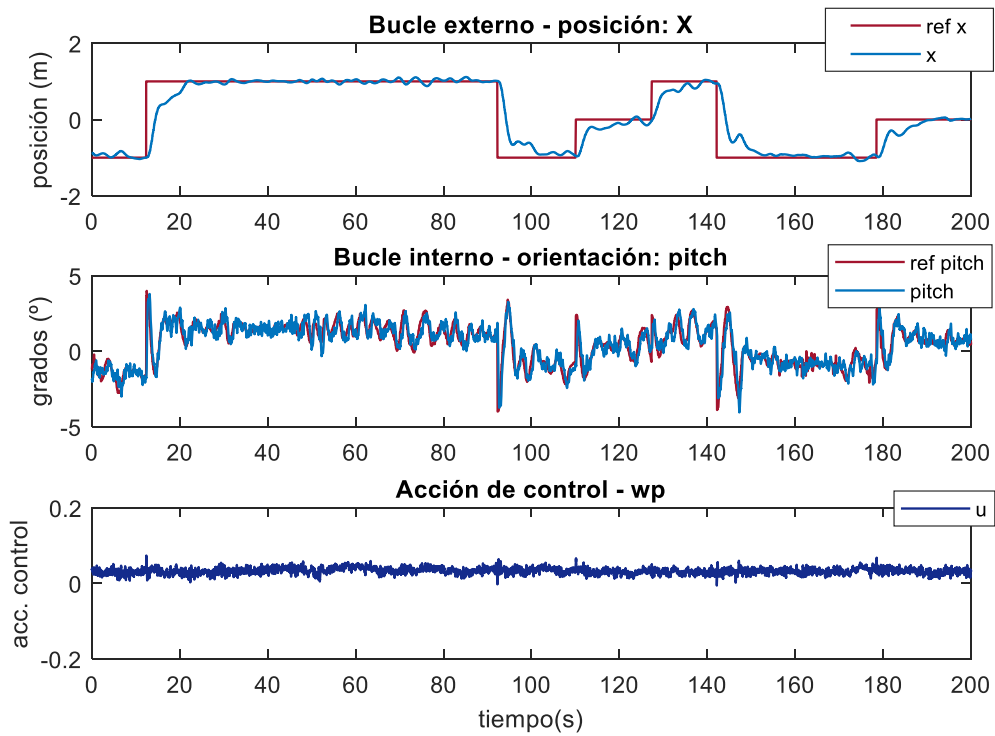


Figura 42 – Gráfica del ensayo en vuelo para el eje X

Control del *eje y* (PD+UDE):

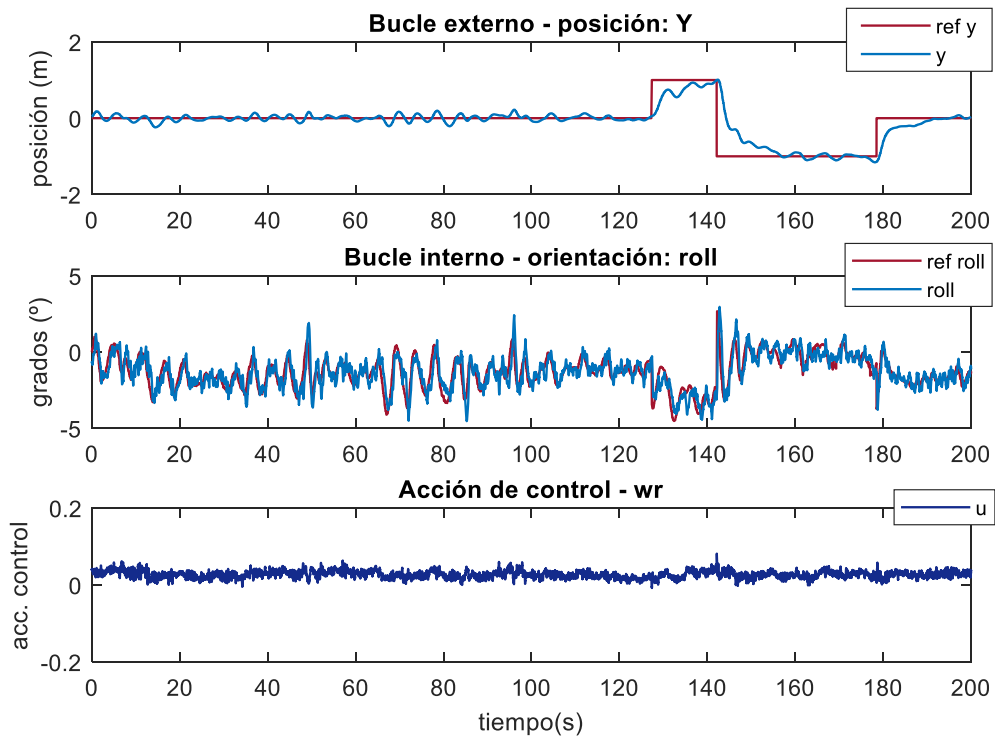


Figura 43 – Gráfica del ensayo en vuelo para el eje Y

Control para inmovilizar el **yaw** (*PD+UDE*):

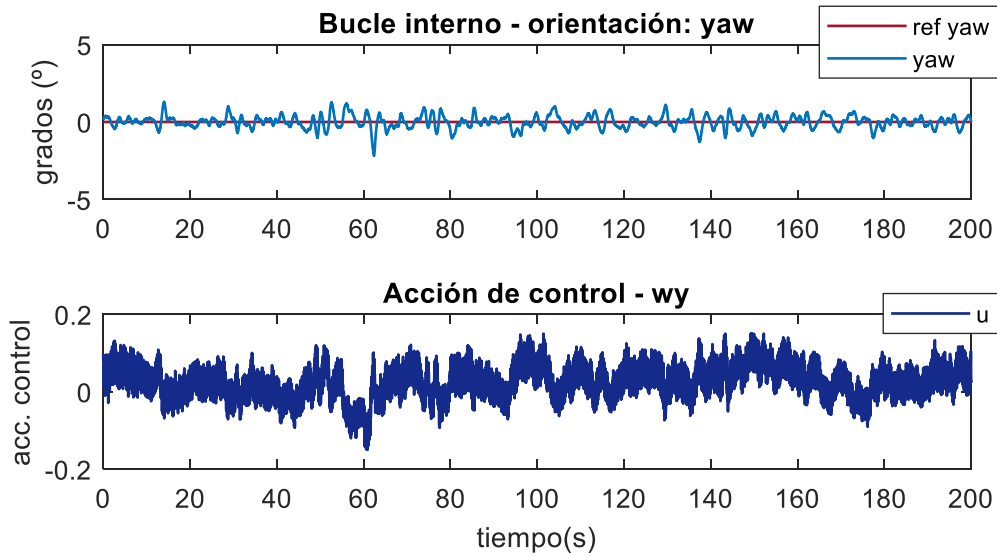


Figura 44 – Gráfica del control del yaw

Como se puede observar (Figura 42 y Figura 43), en los ejes *x* e *y* el dron es capaz de mantener su posición y seguir la referencia marcada con un tiempo de establecimiento de aproximadamente 8 segundos y errores del orden de centímetros. Es capaz de lograr esto a pesar de las turbulencias generadas por el efecto suelo y por las paredes, además de los errores de modelado inducidos por la distribución desequilibrada del peso en el aparato y los pequeños retardos generados por la recepción de las mediciones del sistema *Optitrack*.

En cuanto al control de altitud (*eje z*), este se ha considerado desacoplado del resto de procesos. Responde también al modelo del doble integrador siendo la acción de control la fuerza resultante en dirección vertical generada por los cuatro propulsores (ver apartado Ecuación del modelo traslacional: del Capítulo 4).

Para este eje no se ha implementado el UDE pues se ha observado una respuesta aceptable utilizando únicamente el PD, por ello en la gráfica (Figura 45) se observa un error en régimen permanente ocasionado por una perturbación constante aplicada sobre el vehículo, su propio peso.

Control del **eje z** (*PD*):

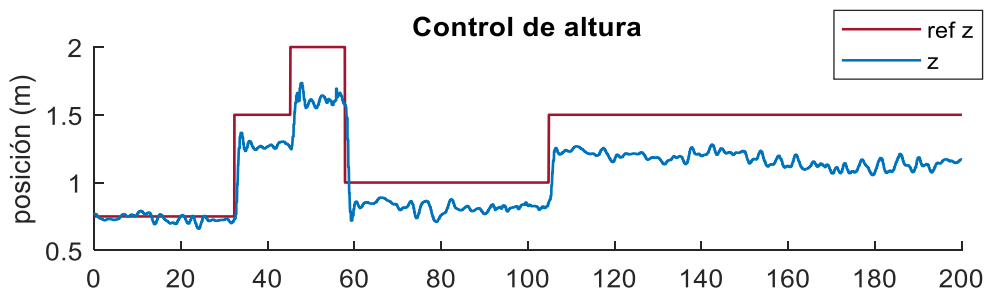


Figura 45 – Gráfica del ensayo en vuelo para el eje Z

Conclusiones

A lo largo de este trabajo de fin de grado se ha hecho un recorrido teórico por diferentes técnicas de control que coexisten en el mercado actual y se ha comprobado su aplicabilidad de forma práctica sobre un vehículo quadrotor destacando sus beneficios e inconvenientes (Capítulo 6).

Se concluye que, pese a que todos los algoritmos ensayados han demostrado ser válidos para el control de la aeronave, las técnicas basadas en estimación del estado se han evidenciado superiores a la acción integral para el control de sistemas inestables, con perturbaciones y de dinámica rápida.

En concreto, para el control de un quadrotor, el *Estimador de incertidumbre y perturbaciones (UDE)* y el *Observador de Estados Extendido (ESO)* manifiestan estas ventajas sin apenas sacrificar la simplicidad y versatilidad que hacen del tradicional control PID el líder del mercado.

El programa *Horizon 2020* [31] de la Comisión Europea destaca el campo de los UAVs como un sector estratégico para potenciar su desarrollo y, finalmente, lograr la total integración de estas aeronaves en el espacio aéreo europeo. Con miras a este futuro, donde se hará necesario garantizar ciertos estándares de seguridad frente a todo tipo de condiciones climatológicas, los resultados arrojados por este proyecto ilustran que es posible introducir nuevas estrategias de control con mejores prestaciones que los convencionales PIDs y posicionan a los algoritmos mencionados como una oportunidad para lograrlo.



Figura 46 – Dron en vuelo hover estabilizado sobre el origen de coordenadas

Como propuestas para futuros trabajos que se realicen en torno a este sistema quadrotor, la evolución lógica para su control de posición sería liberar el *yaw* e implementar una matriz de rotación sobre las medidas del *Optitrack* permitiendo de este modo el control de la ubicación del quadrotor con cualquier valor del *yaw*.

Además, podría ser interesante añadir un filtro de Kalman a las medidas que utilizase las mediciones del *Optitrack* y las del acelerómetro de la IMU para minimizar de este modo los retardos que sufre este sistema de mediciones.

Finalmente, para lograr un dron plenamente operativo, sustituir las medidas del *Optitrack* por un GPS y preparar el vehículo para el vuelo exterior.

Respecto a esto último que se menciona, se ha de destacar que los experimentos realizados hasta la fecha traslucen que el principal inconveniente para la viabilidad de los UAV es, sin lugar a duda, la tecnología de las baterías del mercado actual y su relación *peso-duración*. Con las baterías disponibles en el laboratorio los vuelos únicamente podían durar escasos minutos, lo que nos ha obligado a emplear una fuente de alimentación para los ensayos.

Este es el mismo reto al que se ha enfrentado hasta ahora la automoción eléctrica, pero en nuestro caso se trata de un inconveniente más crítico si cabe y será el progreso en esta área lo que realmente abra las puertas al futuro deseado por la Comisión Europea.

Anexo I – Simulaciones programadas

Como herramienta para facilitar el diseño de los controladores se han creado en MATLAB® diversos procesos en bucle cerrado que implementan cada una de las técnicas con que se ha experimentado a lo largo de este trabajo de fin de grado.

Además, también se han creado scripts para automatizar la obtención de datos de los experimentos, su comparación con el modelo diseñado, el ajuste de dicho modelo y la obtención de gráficas.

Subsistema de rotación

Este proceso es el que se ha estudiado en más profundidad, dado que constituye la base del control de un cuadricóptero.

Como aproximación inicial a cada una de las técnicas de control se han creado simulaciones del proceso en bucle cerrado implementando en este las funciones de transferencia continuas de los algoritmos. De este modo se ha podido experimentar directamente con el ajuste realizado en *rltool* y se ha obtenido una comprensión teórica más profunda de su funcionamiento.

Se ha trabajado empleando funciones de transferencia introducidas desde el *workspace* de MATLAB® a través del bloque *LTI System* de *Simulink*.

A continuación, se mostrarán los esquemas de las simulaciones realizadas.

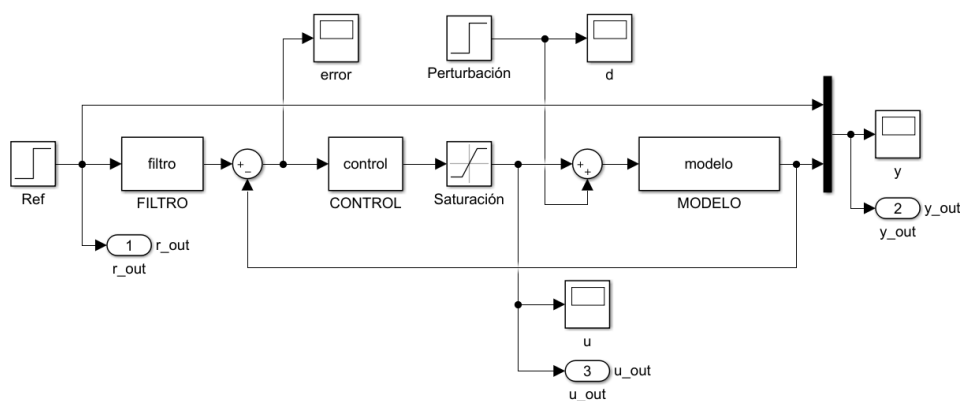


Figura 47 – Simulación del control PID con filtro en la referencia (continuo)

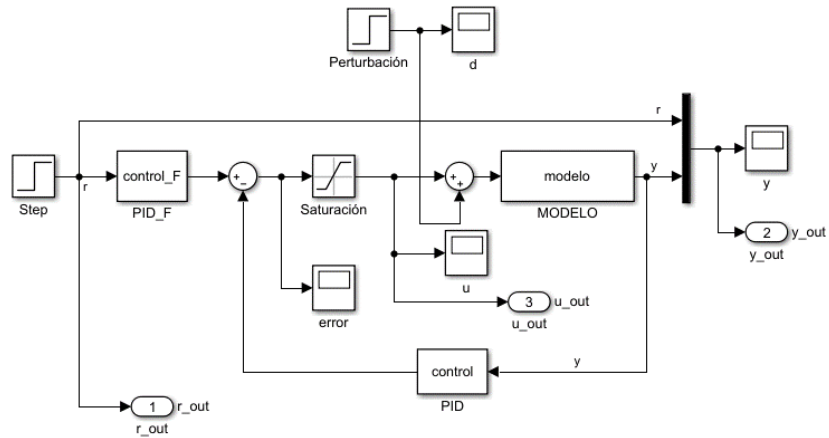


Figura 48 – Simulación del control PID con dos grados de libertad (continuo)

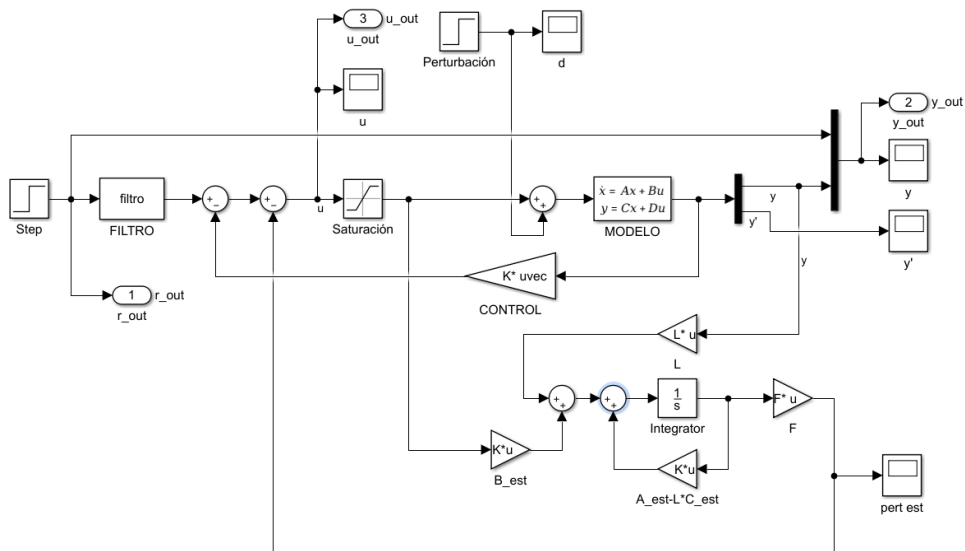


Figura 49 – Simulación del control PD+ESO (continuo)

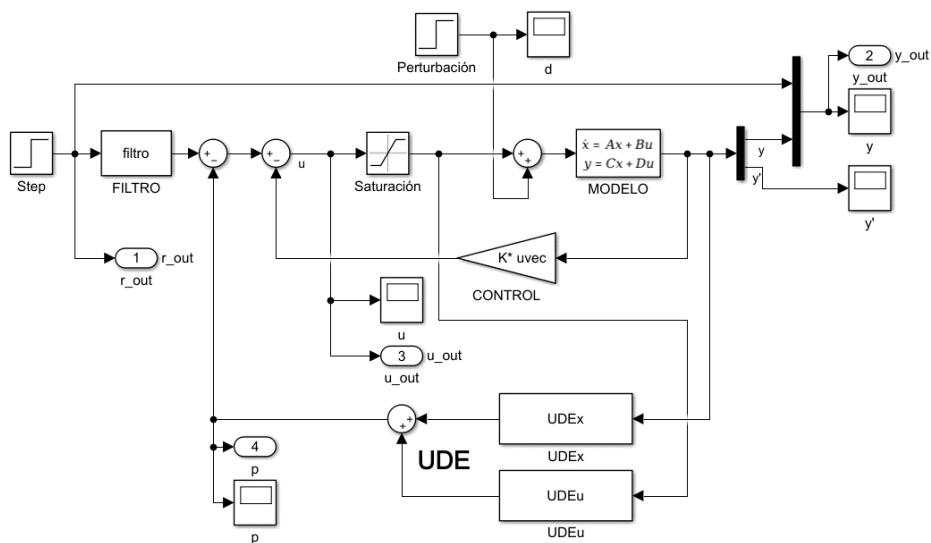


Figura 50 – Simulación del control PD+UDE (continuo)

Como segundo paso, para detectar errores a la hora de discretizar y programar los algoritmos anteriores, se ha diseñado una simulación que implementa el control en forma de código utilizando funciones en el workspace de MATLAB® y el bloque *Interpreted MATLAB Fcn* de *Simulink* junto con un *Zero-Order Hold* de 4ms (periodo de ejecución del código del dron).

Dicha simulación se ejecutaba en un *script* que automatizaba la obtención de gráficas y las solapaba con las adquiridas de los ensayos sobre el vehículo.

Las gráficas obtenidas por esta última simulación son las que acompañan a las mediciones de los ensayos reales en el *capítulo 6 - Sistema de control de la orientación*.

Para la discretización de los controles de acuerdo con el tiempo de muestreo de nuestro equipo nos hemos apoyado en las funciones *c2d* y *d2c* de MATLAB®.

El esquema es el que sigue:

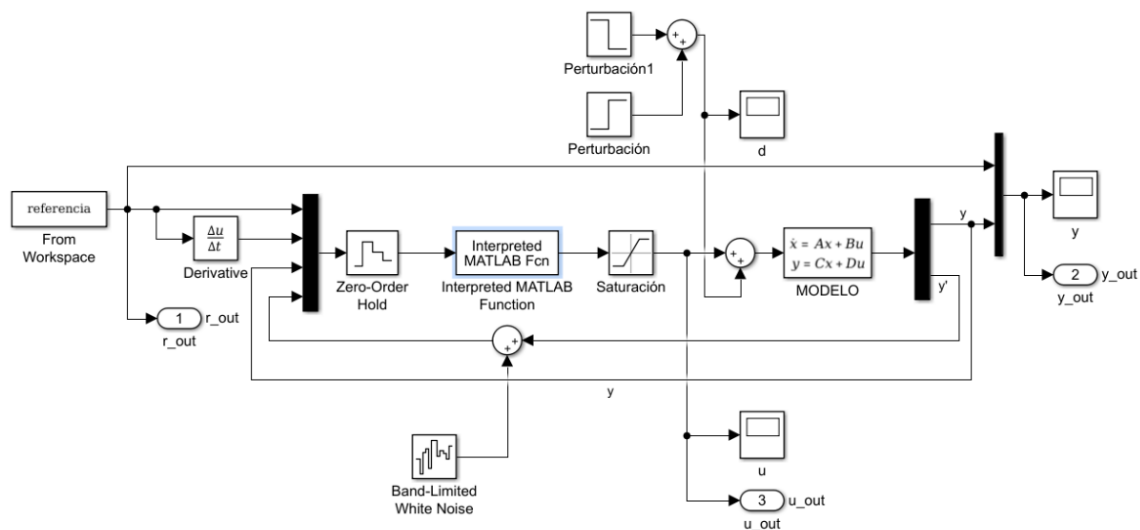


Figura 51 – Simulación del control implementado en código (discreto)

Subsistema de traslación

Para verificar que el control de posición era realizable con un tiempo de muestreo de 200ms y realizar un ajuste inicial de los parámetros, se ha ampliado la estructura mostrada en la Figura 51 añadiendo el modelo del sistema traslacional y el control de este en cascada sobre el anterior.

El control del bucle interno funciona con un periodo de 4ms mientras que el bucle externo utiliza, como ya se ha mencionado, un periodo 25 veces mayor.

A continuación, se adjunta una imagen del esquema del proceso:

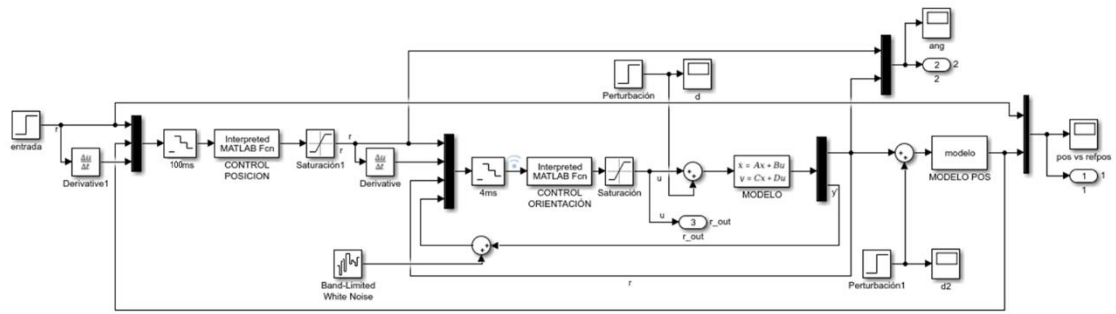


Figura 52 – Simulación del control de los dos subsistemas en cascada (discreto)

Parte 2:

PRESUPUESTO

1. Introducción

Este presupuesto forma parte del trabajo de fin de grado “Diseño, implementación y evaluación de algoritmos de control, para mejorar la estabilidad de un sistema quadrotor con incertidumbre de modelado y perturbaciones externas” de la titulación de Grado en Ingeniería de las Tecnologías Industriales cursada en la Escuela Técnica Superior de Ingenieros Industriales de la Universidad Politécnica de Valencia durante los años 2014-2018.

Este documento tiene como objetivo proporcionar toda la información sobre los costes que ha supuesto la ejecución del proyecto mencionado. Se detallará toda la información sobre los gastos en las distintas actividades realizadas, el tiempo de desarrollo, los salarios y el coste de la maquinaria empleada.

Para el cálculo de los costes directos complementarios, los costes indirectos, los gastos generales y el beneficio industrial se han utilizado los valores porcentuales estándar recomendados para proyectos de estas características.

2. Cuadro de precios de mano de obra

Las personas que han trabajado en el proyecto son un un graduado en ingeniería en tecnologías industriales y un técnico que supervisaba y asistía con el uso del material del laboratorio. Su salario ha sido obtenido realizando la media entre los que figuraban en los distintos proyectos consultados y el cálculo de las horas dedicadas se ha realizado de forma aproximada.

Tabla 8 – Cuadro de precios de mano de obra

| Código | Unidades | Descripción | Precio (€) |
|-----------------|-----------------|--|-------------------|
| MO. GITI | h | Graduado en Ingeniería en Tecnologías Industriales | 20,00 |
| MO. TEC | h | Técnico | 15,00 |

3. Cuadro de precios de maquinaria

Buscando la mayor exactitud posible, el precio del uso del instrumental del laboratorio se ha calculado como el precio total del artículo dividido entre su vida útil obteniendo su precio por hora. Para ello nos hemos basado en presupuestos de anteriores proyectos realizados en el mismo laboratorio [6], [13], [28], [15], [5].

Se ha supuesto que la amortización de los programas con licencia no se realiza únicamente en este proyecto, sino que son consumidos también en tareas paralelas.

Para la estimación de la vida útil del equipo se han tomado tres estándares:

- Duración de un año: 250 (días laborables en 2018) · 6h/día = 1500h.
- Vida *media*: 3000h
- Vida *media-larga*: 6000h

Tabla 9 – Cuadro de precios de la maquinaria

| Código | Unidades | Descripción | Precio total (€) | Vida útil (h) | Precio (€) |
|----------------|-----------------|---------------------------|-------------------------|----------------------|-------------------|
| M. QUAD | h | Quadrotor | 4808,61 | 3000 | 1,60 |
| M. ROT | h | Banco de ensayos | 1395,47 | 6000 | 0,23 |
| M. OPT | h | Sistema Optitrack | 1750,00 | 6000 | 0,29 |
| M. MOT | h | Licencia de uso Motive | 800,00 | 1500 | 0,53 |
| M. MAT | h | Licencia MATLAB | 2000,00 | 1500 | 1,33 |
| M. PC | h | Ordenador portatil | 1000,00 | 6000 | 0,17 |
| M. PC2 | h | Ordenador del laboratorio | 500,00 | 6000 | 0,08 |

4. Cuadro de precios unitarios

Tabla 10 – Cuadro de precios unitarios

| Código | Unidades | Descripción | Rdto. | Precio (€) | Importe (€) |
|---------------|-----------------|--|--------------|-------------------|--------------------|
| UO 1 | ud | Diseño de las simulaciones | | 133,66 | |
| | | Costes Directos | 1 | 128,52 | 128,52 |
| | % | Costes Indirectos | 0,04 | 128,52 | 5,14 |
| UO 2 | ud | Identificación del sistema | | 167,61 | |
| | | Costes Directos | 1 | 161,17 | 161,17 |
| | % | Costes Indirectos | 0,04 | 161,17 | 6,45 |
| UO 3 | ud | Diseño de algoritmos de control | | 132,07 | |
| | | Costes Directos | 1 | 126,99 | 126,99 |
| | % | Costes Indirectos | 0,04 | 126,99 | 5,08 |
| UO 4 | ud | Ajuste y validación de algoritmos de control de orientación | | 81,33 | |
| | | Costes Directos | 1 | 78,20 | 78,20 |
| | % | Costes Indirectos | 0,04 | 78,20 | 3,13 |

| Código | Unidades | Descripción | Rdto. | Precio (€) | Importe (€) |
|---------------|-----------------|---|--------------|-------------------|--------------------|
| UO 5 | ud | Validación del control del dron en vuelo | | 201,99 | |
| | | Costes Directos | 1 | 194,22 | 194,22 |
| | % | Costes Indirectos | 0,04 | 194,22 | 7,77 |

5. Cuadro de precios descompuestos

Tabla 11 – Cuadro de precios descompuestos

| Código | Unidades | Descripción | Rdto. | Precio (€) | Importe (€) |
|---------------|-----------------|--|--------------|-------------------|--------------------|
| UO 1 | ud | Diseño de las simulaciones | | 128,52 | |
| MO GITI | h | Graduado en Ingeniería en tecnologías Industriales | 6 | 20,00 | 120,00 |
| M. PC1 | h | Ordenador portátil | 4 | 0,17 | 0,67 |
| M. MAT | h | Licencia MATLAB | 4 | 1,33 | 5,33 |
| | % | Costes Directos Complementarios | 0,02 | 126,00 | 2,52 |
| UO 2 | ud | Identificación del sistema | | 161,17 | |
| MO GITI | h | Graduado en Ingeniería en tecnologías Industriales | 5 | 20,00 | 100,00 |
| MO. TEC | h | Técnico | 3 | 15,00 | 45,00 |
| M. PC1 | h | Ordenador portátil | 5 | 0,17 | 0,83 |
| M. MAT | h | Licencia MATLAB | 5 | 1,33 | 6,67 |
| M. QUAD | h | Quadrotor | 3 | 1,60 | 4,81 |
| M. ROT | h | Banco de ensayos | 3 | 0,23 | 0,70 |
| | % | Costes Directos Complementarios | 0,02 | 158,01 | 3,16 |

| <i>Código</i> | <i>Unidades</i> | <i>Descripción</i> | <i>Rdto.</i> | <i>Precio (€)</i> | <i>Importe (€)</i> |
|---------------|-----------------|--|--------------|-------------------|--------------------|
| UO 3 | ud | Diseño de algoritmos de control | | 126,99 | |
| MO GITI | h | Graduado en Ingeniería en tecnologías Industriales | 6 | 20,00 | 120,00 |
| M. PC1 | h | Ordenador portátil | 3 | 0,17 | 0,50 |
| M. MAT | h | Licencia MATLAB | 3 | 1,33 | 4,00 |
| | % | Costes Directos Complementarios | 0,02 | 124,50 | 2,49 |
| UO 4 | ud | Ajuste y validación de algoritmos de control de orientación | | 78,20 | |
| MO GITI | h | Graduado en Ingeniería en tecnologías Industriales | 2 | 20,00 | 40,00 |
| MO. TEC | h | Técnico | 2 | 15,00 | 30,00 |
| M. PC1 | h | Ordenador portátil | 2 | 0,17 | 0,33 |
| M. MAT | h | Licencia MATLAB | 2 | 1,33 | 2,67 |
| M. QUAD | h | Quadrotor | 2 | 1,60 | 3,21 |
| M. ROT | h | Banco de ensayos | 2 | 0,23 | 0,47 |
| | % | Costes Directos Complementarios | 0,02 | 76,67 | 1,53 |
| UO 5 | ud | Validación del control del dron en vuelo | | 194,22 | |
| MO GITI | h | Graduado en Ingeniería en tecnologías Industriales | 5 | 20 | 100,00 |
| MO. TEC | h | Técnico | 5 | 15 | 75,00 |
| M. QUAD | h | Quadrotor | 5 | 1,60 | 8,01 |
| M. OPT | h | Sistema Optitrack | 5 | 0,29 | 1,46 |
| M. MOT | h | Licencia de uso Motive | 5 | 0,53 | 2,67 |
| M. MAT | h | Licencia MATLAB | 5 | 1,33 | 6,67 |
| M. PC2 | h | Ordenador del laboratorio | 5 | 0,08 | 0,42 |

6. Presupuesto Base de Licitación

Tabla 12 – Presupuesto Base de Licitación

| Código | Unidades | Descripción | Rdto. | Precio (€) | Importe (€) |
|--|-----------------|---|--------------|-------------------|--------------------|
| UO 1 | 801,96 | Diseño de las simulaciones | 6 | 133,38 | 800,27 |
| UO 2 | 167,61 | Identificación del sistema | 1 | 117,19 | 117,19 |
| UO 3 | 792,42 | Diseño de algoritmos de control | 6 | 131,86 | 791,14 |
| UO 4 | 406,66 | Ajuste y validación de algoritmos de control de orientación | 5 | 47,81 | 239,05 |
| UO 5 | 201,99 | Validación del control del dron en vuelo | 1 | 197,38 | 197,38 |
| Presupuesto de Ejecución Material (PEM) | | | | | 2370,65 |
| <i>Gastos Generales (13% PEM)</i> | | | | | 308,18 |
| <i>Beneficio Industrial (6% PEM)</i> | | | | | 142,24 |
| Presupuesto de Ejecución por Contrata (PEC) | | | | | 2821,07 |
| <i>IVA (21% PEC)</i> | | | | | 592,43 |
| Presupuesto Base de Licitación | | | | | 3413,50 |

El presupuesto asciende a la cantidad de TRES MIL CUATROCIENTOS TRECE euros con CINCUENTA céntimos.

Bibliografía

- [1] CASTILLO, P., LOZANO, R., & DZUL, A. E. (2006). *Modelling and control of mini-flying machines* (pp. 39-59). Physica-Verlag.
- [2] MELERO CAZORLA, D. (2013). *Modelo dinámico y diseño de estrategia de control mediante estimadores para el vuelo autónomo de un quadrotor*. Proyecto Final de Carrera. Almería: Universidad de Almería.
- [3] (Página Web) *De Bothezat*.
<http://www.aviastar.org/helicopters_eng/bothezat.php>.
- [4] (Página Web) *Juan de la Cierva*.
<<https://www.biografiasyvidas.com/biografia/c/cierva.htm>>.
- [5] SANZ DÍAZ, R. (2015). *Improving attitude estimation and control of quadrotor systems*. Trabajo Final de Máster. Valencia: Universidad Politécnica de Valencia.
- [6] FRASQUET, A. C. (2015). *Desarrollo integral de un QuadRotor: Diseño de un algoritmo de control para la posición xy basado en señales GPS*. Proyecto Final de Carrera. Valencia: Universidad Politécnica de Valencia.
- [7] OGATA, K. (2003). *Ingeniería de control moderna*. Pearson Educación.
- [8] RUBIO, F. R., & SÁNCHEZ, M. J. L. (1996). *Control adaptativo y robusto* (Vol. 9). Sevilla: Universidad de Sevilla.
- [9] VARGAS, A. M., GÓMEZ, G. R., & CARRANZA, J. M. (2017). *Quadrotor Flight in Constrained Indoor Environments*. Puebla, Mexico.
- [10] LIM, H., PARK, J., LEE, D., & KIM, H. J. (2012). Build your own quadrotor: Open-source projects on unmanned aerial vehicles. *IEEE Robotics & Automation Magazine*, 19(3), 33-45.
- [11] (Página Web) *Amazon Prime air*. <<https://www.amazon.com/Amazon-Prime-Air/b?ie=UTF8&node=8037720011>>.
- [12] España. Real Decreto 1036/2017, de 15 de diciembre, por el que se regula la utilización civil de las aeronaves pilotadas por control remoto. *Boletín Oficial del Estado*, 15 de diciembre de 2017, núm. 316, pp. 129609 a 129641 .
- [13] GARÍN, V. B. (2017). *Diseño e implementación de una plataforma para la validación segura de algoritmos de control avanzados en vehículos aéreos no tripulados tipo quadrotors*. Trabajo Final de Máster. Valencia: Universidad Politécnica de Valencia.
- [14] (Página Web) *Sitio oficial de Raspberry Pi*. <<https://www.raspberrypi.org/>>.

- [15] GARÍN, V. B. (2015). *Sistema de Tiempo Real basado en Raspberry PI para el control de un Quadrotor*. Trabajo Final de Grado. Valencia: Universidad Politécnica de Valencia.
- [16] VIGOUROUX CAVOLINA, D. (2010). *Implementación de unidad de mediciones inerciales (IMU) para robótica utilizando filtro de Kalman*. Caracas: Universidad de Simón Bolívar.
- [17] EUSTON, Mark, et al. (2008). A complementary filter for attitude estimation of a fixed-wing UAV. En *Intelligent Robots and Systems, 2008. IROS 2008. IEEE/RSJ International Conference on*. IEEE, 2008. p. 340-345.
- [18] BOUFFARD, J. L. (2016). *An Alternative Sensor Fusion Method For Object Orientation Using Low-Cost Mems Inertial Sensors*. Trabajo de Fin de Máster. University of Vermont.
- [19] (Página Web) *Sitio oficial de Code::Blocks*. <<http://www.codeblocks.org/>>.
- [20] (Página Web) *Sitio oficial de PuTTY*. <<https://www.putty.org/>>.
- [21] (Página Web) *Sitio oficial de WinSCP*. <<https://winscp.net/eng/download.php>>.
- [22] (Página Web) *Sitio oficial de Optitrack*. <<http://optitrack.com/>>.
- [23] (Página Web) *Cámara Optitrack - Modelo Flex:V100*. <<http://optitrack.com/support/hardware/flex-v100.html>>.
- [24] CASTILLO, P., & LOZANO, R. *Dynamic model of a quad-rotor rotorcraft*. Heudiasyc Laboratory, UTC-CNRS 6599, Compiègne, France
- [25] CASTILLO-GARCIA, P., HERNANDEZ, L. E. M., & GIL, P. G. (2016). *Indoor Navigation Strategies for Aerial Autonomous Systems*. Butterworth-Heinemann.
- [26] MAZZONE, V. (2002). *Controladores PID*. Quilmes: Universidad Nacional de Quilmes.
- [27] LUENBERGER, D. (1971). *An introduction to observers*. IEEE Transactions on automatic control, 16(6), 596-602.
- [28] FRASQUET, A. C. (2016). *Desarrollo de un sistema de control de la orientación de un Quadrotor basado en Cuaterniones y en una técnica de estimación de la incertidumbre de modelado*. Trabajo Final de Máster. Valencia: Universidad Politécnica de Valencia.
- [29] SANZ, Ricardo, et al. Robust control of quadrotors based on an uncertainty and disturbance estimator. *Journal of Dynamic Systems, Measurement, and Control*, 2016, vol. 138, no 7, p. 071006.
- [30] (Página Web) *Sitio oficial de MathWorks*. <https://es.mathworks.com/?s_tid=gn_logo>.
- [31] (Página Web) *Sitio oficial del programa "Horizon 2020" de la CE*. <<https://ec.europa.eu/programmes/horizon2020/>>.

- [32] (Página Web) TESAUROS, *Término: Nitrato de Celulosa*.
<<http://tesauros.mecd.es/tesauros/materias/1029929.html>>.
- [33] (Página Web) *Sitio oficial de Preempt_RT*.
<https://rt.wiki.kernel.org/index.php/Main_Page>.
- [34] (Página Web) *History Lists: Leonardo da Vinci*.
<<http://historylists.org/other/9-incredible-leonardo-da-vinci-inventions.html>>.
- [35] (Página Web) *Cultural China: Bamboo Dragonflies*.
<<https://web.archive.org/web/20101220020133/http://cultural-china.com/chinaWH/html/en/11Kaleidoscope1053.html>>.

