

CENTRO DE INVESTIGACIÓN Y DE ESTUDIOS  
AVANZADOS DEL INSTITUTO POLITÉCNICO  
NACIONAL

UNIDAD ZACATENCO

DEPARTAMENTO DE INGENIERÍA ELÉCTRICA

SECCIÓN DE MECATRÓNICA

**Control de posición de un objeto pendiente de un  
cuadrirotor en el plano vertical**

**T E S I S**

Que presenta

**AARÓN JOSAFAT HERNÁNDEZ MEJÍA**

Para obtener el grado de

**MAESTRO EN CIENCIAS  
EN LA ESPECIALIDAD DE  
INGENIERÍA ELÉCTRICA**

Director de la Tesis

**DR. EDUARDO ARANDA-BRICAIRE**

México, D.F.

NOVIEMBRE, 2015



# Agradecimientos

**A mis padres:** Por su confianza en mí y sus palabras de aliento para alcanzar mis metas. Por todo su esfuerzo para asegurarme una buena educación y nivel de vida. Trabajaré duro y con empeño para enorgullecerlos donde quiera que estén.

**Al Dr. Eduardo Aranda:** Por sus consejos, apoyo y críticas para la culminación de este trabajo. Por darme la oportunidad de trabajar con él y por confiar en mí para el desarrollo de este proyecto de tesis.

**A los doctores de la sección de mecatrónica:** Por su enseñanza y sus consejos. Por haberme dado la oportunidad de realizar mis estudios de maestría en la sección de mecatrónica.

**A Luis Ángel Cuevas:** Por haber estado presente desde el principio hasta el final.

**A mis amigos de la sección:** Por compartir su amistad conmigo y por los buenos momentos que pasamos juntos.

**Al CONACYT:** Por su apoyo económico para realizar mis estudios de maestría.



# Abstract

*In this Master Thesis the development of a control strategy that makes an object hanged from an Unmanned Aerial Vehicle (UAM) be able to track a desired trajectory is approached. It either deals with the hanged object stabilization over a point on the space (regulation) or controlling its position throughout a desired trajectory (trajectory tracking) problems. In both cases, the control strategy is able to rectify and reverse (counteract) the effects of perturbations which are present during operation, such as hanged object deviation due to stream of air, among others.*

*The design of this automatic control system is a class of hierarchical control law composed of two stages.*

*At the first of these it is supposed that UAV and hanged object are inert masses whitout self control and UAV mass could be strategically displaced over space through thrust forces, in such a way that hanged object may be driven in the desired direction. The thrust forces acting on the UAV mass are calculated at this stage through a suitable control law.*

*At the second stage it is known that aerial vehicule used in this thesis already has an internal controller of its manufacturer, and allows the user to drive it through wireless control signals from a portable device or workstation. Hence, it is designed a new controller of the proportional-integral-derivative (PID) type which is able to determinate suitable control signals sended to UAV in order to make its displacement to be as if UAV were pushed by trust forces calculated at first stage.*

*The both control laws, created at the previous stages, work together in a hierarchical way in order to either deal with the hanged object stabilization over a point on the space or controlling its position throughout a desired trajectory.*

*These control laws requires knowledge of some physical variables of the system (such as orientation angles and velocities, as well as cartesian positions, velocities and accelerations, etc.) known as state variables, whose measurement becomes a high technical difficulty. Hence, it is also proposed the design of a mathematical object that builds approximations of these state variables.*

*In order to check for this controller performance, some numerical simulations and real-time experiments have been conducted. Results obtained in both cases were satisfactory even when some disturbances were intentionally produced on the system.*

*Finally, the results of the experiments are shown and discussed.*



# Resumen

En este trabajo de tesis se propone el desarrollo de una estrategia de control que permita el seguimiento de trayectorias de un objeto suspendido en el aire a través de una cuerda desde un vehículo aéreo no tripulado (UAV por sus siglas en inglés). Se aborda el problema de estabilizar el objeto pendiente en un punto en el espacio (regulación), o bien, controlarlo a lo largo de una trayectoria deseada (seguimiento de trayectoria). En ambos casos, la estrategia de control es capaz de corregir y contrarrestar el efecto de las perturbaciones presentes durante la operación; tales como la desviación del objeto debido a corrientes de aire, entre otras.

El proceso de diseño del sistema de control automático aquí planteado es una clase de control jerárquico que está compuesto por dos etapas.

En la primera de ellas se considera que el UAV y objeto pendiente son masas inertes sin control propio y que la masa del UAV podría ser desplazada estratégicamente en el espacio mediante fuerzas de empuje, de tal modo que la masa pendiente sea conducida en la dirección deseada. Las fuerzas de empuje que actúan sobre la masa del UAV son calculadas en esta etapa mediante una ley de control apropiada.

En la segunda etapa se considera que el vehículo aéreo que se utiliza en esta tesis ya posee un control interno de su fabricante, dejando al usuario la posibilidad de dirigirlo mediante señales de control transmitidas inalámbricamente desde un dispositivo móvil o una estación de trabajo fija. Entonces, se diseña una nueva ley de control tipo proporcional integral derivativo (PID) que determine las señales de control adecuadas hacia el UAV que logren que su desplazamiento sea el equivalente al que producirían las fuerzas de empuje obtenidas en la primera etapa.

Ambas leyes de control, creadas en las dos etapas mencionadas, funcionan conjuntamente de manera jerárquica a fin de lograr el seguimiento de una trayectoria deseada para el objeto pendiente, o bien, su regulación en un punto en el espacio.

Dichas leyes de control requieren del conocimiento de algunas variables físicas del sistema (tales como ángulos y velocidades de orientación, así como también posiciones, velocidades y aceleraciones cartesianas, etc.) conocidas como variables de estado, cuya medición representa una fuerte dificultad técnica. Por lo anterior, se propone también el diseño de un objeto matemático que construya una aproximación de dichas variables de estado.

Finalmente, se muestran y discuten los resultados obtenidos en experimentos en simulación numérica y tiempo real.



# Índice general

<b>1. Introducción</b>	<b>3</b>
1.1. Antecedentes y motivación . . . . .	3
1.2. Planteamiento del problema . . . . .	4
1.3. Objetivos generales y específicos del proyecto . . . . .	5
<b>2. Descripción del Prototipo Experimental</b>	<b>7</b>
2.1. El AR-Drone de Parrot . . . . .	7
2.2. Sistema óptico de captura de movimiento OptiTrack . . . . .	10
2.3. Robot Operating System (ROS) . . . . .	11
2.3.1. Controlador del AR-Drone . . . . .	13
2.3.2. Estrategia de comunicación . . . . .	14
<b>3. Modelado y control del sistema</b>	<b>17</b>
3.1. Cuadrirotor de configuración “tipo X” . . . . .	17
3.2. Caracterización del AR-Drone . . . . .	20
3.3. Sistema cuadrirotor-péndulo . . . . .	24
<b>4. Estimación de estados del sistema</b>	<b>35</b>
4.1. Diferenciador de Levant . . . . .	37
<b>5. Resultados en simulación y tiempo real</b>	<b>41</b>
5.1. Experimentos en simulación . . . . .	41
5.1.1. Regulación del objeto pendiente en un punto en el plano . . .	42
5.1.2. Seguimiento de trayectoria . . . . .	42
5.2. Experimentos en tiempo real . . . . .	45
5.2.1. Regulación del objeto pendiente en un punto en el plano . . .	46
5.2.2. Seguimiento de trayectoria . . . . .	47
<b>6. Conclusiones</b>	<b>53</b>
<b>Bibliografía</b>	<b>57</b>
<b>A. Artículo publicado</b>	<b>59</b>



# Capítulo 1

## Introducción

Este capítulo describe los antecedentes y motivaciones para desarrollar este trabajo de tesis. Abarca los objetivos generales y específicos, el planteamiento del problema y la hipótesis de trabajo que dieron origen al presente desarrollo.

### 1.1. Antecedentes y motivación

El pionero francés en aviación Etienne Oehmichen comprobó que era posible la construcción del cuadricóptero con la fabricación del Oehmichen No. 2 en 1922. Dicha aeronave es el primer antecedente histórico de un cuadricóptero práctico. En la actualidad, los cuadricópteros, también llamados *drones*, son comúnmente diseñados como vehículos no tripulados, por lo tanto, son de pequeñas dimensiones y pueden ser conducidos en interiores y exteriores. Los avances en tecnología de semiconductores y MEMs<sup>1</sup> han hecho posible el desarrollo de pequeñas IMUs<sup>2</sup>, útiles para estimar la orientación del vehículo aéreo en el espacio tridimensional. Dichas IMUs frecuentemente están incorporadas en las tarjetas de circuito impreso de los controladores a bordo de los vehículos.

En los vehículos no tripulados, también denominados *drones*, el control se realiza de manera remota haciendo uso de alguna tecnología inalámbrica como radiofrecuencia, WiFi, etc. El operador toma el control de la aeronave a través de una interfaz, interpreta la información disponible y actúa en consecuencia. De lo expuesto, sobresalen tres puntos:

1. La navegación del vehículo aéreo no es completamente autónoma, sin embargo, se han propuesto varias soluciones clasificadas como *Simultaneous Localization And Mapping* (SLAM) que ya han sido desarrolladas para robots terrestres.
2. El vehículo aéreo no tiene conocimiento del ambiente que le rodea. Toda la información generada por sus sensores es enviada a la base de control para su

---

<sup>1</sup>Acrónimo en inglés: *Micro Electrical Mechanisms*, dispositivos micro electromecánicos.

<sup>2</sup>Acrónimo en inglés: *Inertial Measurement Unit*, unidad de medición inercial.

interpretación.

3. El usuario requiere una interfaz para controlar al dron. Las interfaces son simples y eficientes, así facilitarán que el usuario cumpla su objetivo. Regularmente se desarrollan tomando como modelo alguna analogía con la que el usuario está familiarizado. Las mejores interfaces para la manipulación remota de aeronaves son de tipo hápticas y se asemejan a una cabina de avión. Cabe mencionar que todas las desiciones son tomadas por el piloto en tiempo de vuelo.

Existen dos escenarios de operación del dron ya desarrollados y reportados. En el primero, el dron es controlado a distancia por el usuario con un dispositivo de comunicación móvil. En este escenario, el usuario está a cargo de todas las maniobras. En el segundo escenario, el dron es controlado por una estación de trabajo fija en tierra donde la información es recibida y procesada por algoritmos de control con algún propósito de navegación.

En este trabajo de tesis se trabajará sobre un escenario similar al segundo. El propósito de navegación del cuadricóptero es controlar la posición de una carga que se encuentra suspendida de él a través de una cuerda. La ubicación de la carga pendiente y del cuadricóptero en el espacio se determina a través de un sistema de cámaras para captura de movimiento. Este sistema de captura de movimiento entrega la información a una unidad central de procesamiento (CPU) donde se realiza el procesamiento de dicha información. Luego, desde el CPU se envía al cuadricóptero una señal de control a fin de ejercer un control indirecto sobre el objeto pendiente y cumplir algún propósito (por ejemplo, su regulación en un punto en el espacio, seguimiento de trayectoria, etc).

## 1.2. Planteamiento del problema

Los procesos tecnológicos avanzados proporcionan nuevas oportunidades en la industria. Es importante desarrollar e implementar tecnología avanzada en los equipos que forman parte de los procesos industriales. Entre los equipos que tienen un uso muy extendido en el manejo de material se encuentran las grúas viajeras; por esta razón es importante automatizarlas y mejorar su desempeño.

Tradicionalmente, las grúas viajeras han sido utilizadas para mover manualmente materiales desde una locación a otra. Generalmente, dichos equipos son terrestres y requieren rieles o carriles de soporte para su desplazamiento. Actualmente existen procesos industriales cuyo escenario de operación va sufriendo modificaciones de acuerdo a las nuevas necesidades de trabajo. Esto genera la necesidad de realizar ajustes físicos en los rieles o carriles para redirigir el desplazamiento de las grúas viajeras así como prolongar o acortar su trayectoria, provocando esfuerzo adicional, gastos de mantenimiento y pérdida de tiempo.

Por otra parte, existe una gran variedad de aplicaciones industriales donde es necesario trasladar cargas en áreas que disponen de espacio libre muy pequeño. Así, el balanceo de la carga provocaría su colisión con otros equipos o personal a su alrededor. Más aún, el balanceo de la carga causaría el derrame del material transportado cuando se trate de materiales líquidos o daños al propio material como es el caso de aquellos que deben evitar ser sacudidos durante su transporte.

Por lo anterior, se requiere implementar una grúa viajera que desplace vía aérea y no necesite rieles o carriles de soporte para su desplazamiento y, además, que ejerza un control de posición automatizado de gran precisión sobre la carga para evitar sus oscilaciones.

### 1.3. Objetivos generales y específicos del proyecto

El objetivo general de este proyecto de tesis es manipular estratégicamente el movimiento aéreo de un cuadrirotor de tal modo que, indirectamente, se pueda ejercer el control de posición adecuado de un objeto que se encuentra suspendido de la aeronave a través de una cuerda, evitando sus oscilaciones y así cumplir algún propósito, por ejemplo, estabilizar el objeto pendiente en un punto en el espacio (regulación), o bien, dirigirlo sobre una trayectoria deseada (seguimiento de trayectoria).

Para dar cumplimiento a dicho objetivo se requiere cumplir los objetivos particulares enumerados a continuación.

1. Establecer una comunicación inalámbrica vía WiFi entre una estación de trabajo fija y un cuadrirotor con control interno de su fabricante. Mediante dicha transmisión inalámbrica de datos se tendrá control total de los movimientos del dron.
2. Implementar un sistema óptico de captura de movimiento que permita obtener información en tiempo real de la posición y orientación del cuadrirotor en el espacio, así como de la posición del objeto pendiente.
3. Diseñar un objeto matemático (observador de estados) que construya una aproximación de las variables físicas no medibles del sistema (tales como velocidades, aceleraciones y sobreaceleraciones de orientación y desplazamiento tanto del dron como del objeto pendiente).
4. Diseñar una ley de control que genere las señales de control adecuadas al cuadrirotor, de tal modo que su movimiento aéreo controle la posición sobre un plano vertical del objeto que se encuentra suspendido de él a través de una cuerda.
5. Analizar el desempeño del control diseñado mediante experimentos tanto en simulación numérica como en tiempo real.



# Capítulo 2

## Descripción del Prototipo Experimental

### 2.1. El AR-Drone de Parrot

Para este trabajo de tesis se usará el cuadricóptero modelo AR-Drone 2.0, desarrollado por la compañía Parrot (Fig. 2.1). Este vehículo aéreo funciona propulsado por cuatro motores eléctricos en configuración cuadricóptero. Cuenta con un microprocesador y una serie de sensores, entre los cuales se incluyen dos cámaras que le permiten captar información del exterior, más un transmisor/receptor WiFi integrado que le permite vincularse a dispositivos que cuenten con los sistemas operativos iOS, Android o Linux. Esto permite controlar al cuadricóptero directamente desde un dispositivo móvil o fijo, desde el cual se reciben las imágenes y datos de telemetría que los sensores están captando.



**Figura 2.1.** AR-Drone, del fabricante Parrot

El pilotaje se encuentra parcialmente asistido por el sistema operativo del drone, que se encarga de corregir altitud, orientación y desplazamiento con base en los datos recibidos por los sensores de a bordo. Esto permite seleccionar diferentes modos de pilotaje con diferente grado de asistencia con base en las necesidades del piloto y a las

condiciones del entorno. El despegue y aterrizaje es vertical y totalmente automático, controlado completamente por el sistema operativo del drone.

A continuación se enlistan las principales especificaciones técnicas del drone:

- Especificaciones físicas
  - Dimensiones con casco:  $52.5 \times 51.5$  cm
  - Peso: 460 g con casco para interiores
- Sistema informático integrado
  - Microprocesador ARM9 RISC de 32 bits @ 468 MHz
  - Memoria DDR SDRAM de 128 MB @ 200 MHz
  - Sistema operativo con núcleo Linux
  - Modem WiFi b/g
- Sistemas de guía inercial
  - Acelerómetro de 3 ejes construido con tecnología MEMs
  - Girómetro de 2 ejes
  - Girómetro de precisión de 1 eje para el control de guiñada
- Motores y energía
  - 4 motores sin escobillas, funcionando a 3500 rpm con potencia de 15 W
  - Batería de ion de litio de 3 celdas, a 1000 mA/hora y voltaje nominal de 11.1 V

El paradigma de uso habitual comprende a un dispositivo móvil capaz de conectarse a una red WiFi y el cuadricóptero. La empresa Parrot, fabricante del cuadricóptero, distribuye la aplicación *AR-Drone Free Flight* como punto de partida, la cual ofrece al usuario la posibilidad de acceder a todos los recursos del drone y volarlo. Como es de suponer, una computadora puede reemplazar al dispositivo móvil en el escenario anteriormente descrito. Esta opción predomina en trabajos de investigación con el AR-Drone, pues la computadora con todos sus recursos ejecuta algoritmos mucho más complejos.

### Comunicación drone-CPU

La comunicación se realiza a través de tres puertos UDP, cada uno con distintos propósitos. Desde el punto de vista del CPU adscrito a la red del cuadricóptero, los puertos son los siguientes:

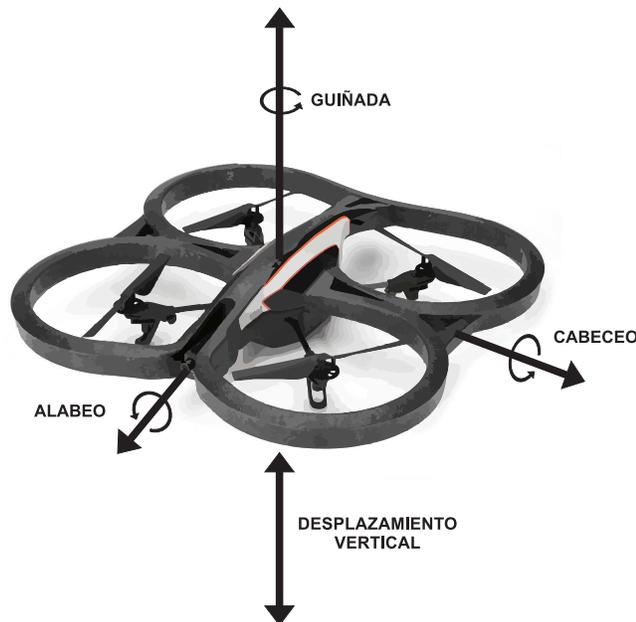
- Puerto UDP 5554: Recepción de los datos de navegación

- Puerto UDP 5555: Recepción del flujo de video
- Puerto UDP 5556: Envío de comandos de control al cuadricóptero

La elección del protocolo UDP es común en este tipo de aplicaciones, pues no es necesario garantizar que no se pierda ningún paquete de información. En estas aplicaciones es mejor enviar el último y más reciente paquete de información a usar recursos para enviar el último paquete que se perdió.

### Control de movimientos del drone

Como se mencionó anteriormente, el pilotaje se encuentra parcialmente asistido por el propio sistema operativo del drone. En la Fig. 2.2 se muestran las cuatro variables sobre las cuales el usuario tiene control directo: ángulo de alabeo, ángulo de cabeceo, velocidad de guiñada y velocidad de desplazamiento vertical.



**Figura 2.2.** Variables controlables del drone

El control del cuadricóptero desde el CPU se realiza a través de comandos enviados en paquetes UDP<sup>1</sup>, puerto 5556. El fabricante recomienda enviar estos paquetes cada

---

<sup>1</sup>UDP es un protocolo “sin conexiones”. A diferencia de TCP, UDP no comprueba los datos que llegan al host de recepción. En lugar de ello, UDP da formato al mensaje que se recibe desde la capa de la aplicación en los paquetes UDP. UDP adjunta un encabezado a cada paquete. El encabezado

30ms para un vuelo suave y con un espaciamiento máximo de 2 segundos para que el drone no dé por perdida la conexión.

Para ello el usuario definirá previamente los ángulos máximos alcanzables de alabeo y cabeceo, así como las velocidades máximas de desplazamiento vertical y de guiñada. Luego, por cada uno de esos argumentos el drone espera recibir como instrucción un valor numérico dentro del rango  $[-1,1]$ , que representa una escala entre el valor máximo alcanzable en un sentido y el mismo valor máximo en sentido contrario, según la variable a controlar.

La guía de desarrollo que ofrece Parrot a través de su página web contiene una mejor descripción de todos los comandos y sus especificaciones.

### Datos de navegación

Toda la información que generan los sensores (ángulos de orientación, altitud del drone respecto al suelo, velocidad de desplazamiento, etc) a bordo del cuadricóptero es enviada a través de UDP, por el puerto 5554. La información está almacenada en formato binario y consiste de varios bloques que el fabricante denomina “opciones”. Cada opción tiene un encabezado para su identificación y la información es almacenada en enteros de 32 bits en formato *little endian*.

La guía de desarrollo que ofrece Parrot a través de su página web contiene la especificaciones técnicas del flujo de datos de navegación.

## 2.2. Sistema óptico de captura de movimiento OptiTrack

Como se mencionó anteriormente, para determinar la ubicación del drone y de la carga pendiente y del cuadricóptero en el espacio de trabajo se hace uso de un sistema de captura de movimiento. Para esta tesis se empleará el Sistema OptiTrack que consiste en un sistema óptico de captura de movimiento tridimensional que utiliza cámaras Flex 13 infrarrojas (Fig. 2.3) y marcadores esféricos reflejantes. Dichos marcadores son colocados sobre el cuadricóptero y el objeto pendiente estratégicamente, de modo que no exista confusión al determinar su orientación en el espacio. Por lo tanto, es conveniente evitar simetrías en la distribución de los marcadores sobre el cuerpo. Se requiere colocar al menos tres marcadores por cada objeto.

Obsérvese la Fig. 2.4. Las cámaras distribuidas en el espacio de trabajo emiten radiación infrarroja que es reflejada por los marcadores y recogida de nuevo en las propias cámaras. Así, las cámaras ofrecen captura de imagen integrada así como seguimiento de movimiento y son capaces de capturar objetos en movimiento rápido con el generador de imágenes obturador global a 100 o 120 FPS de velocidad de

---

contiene los puertos de envío y recepción, un campo con la longitud del paquete y una suma de comprobación.



Figura 2.3. Cámaras Flex 13 del sistema OptiTrack

captura.

El sistema OptiTrack envía y recibe señales de sincronización a través del cable USB a una computadora.

## 2.3. Robot Operating System (ROS)

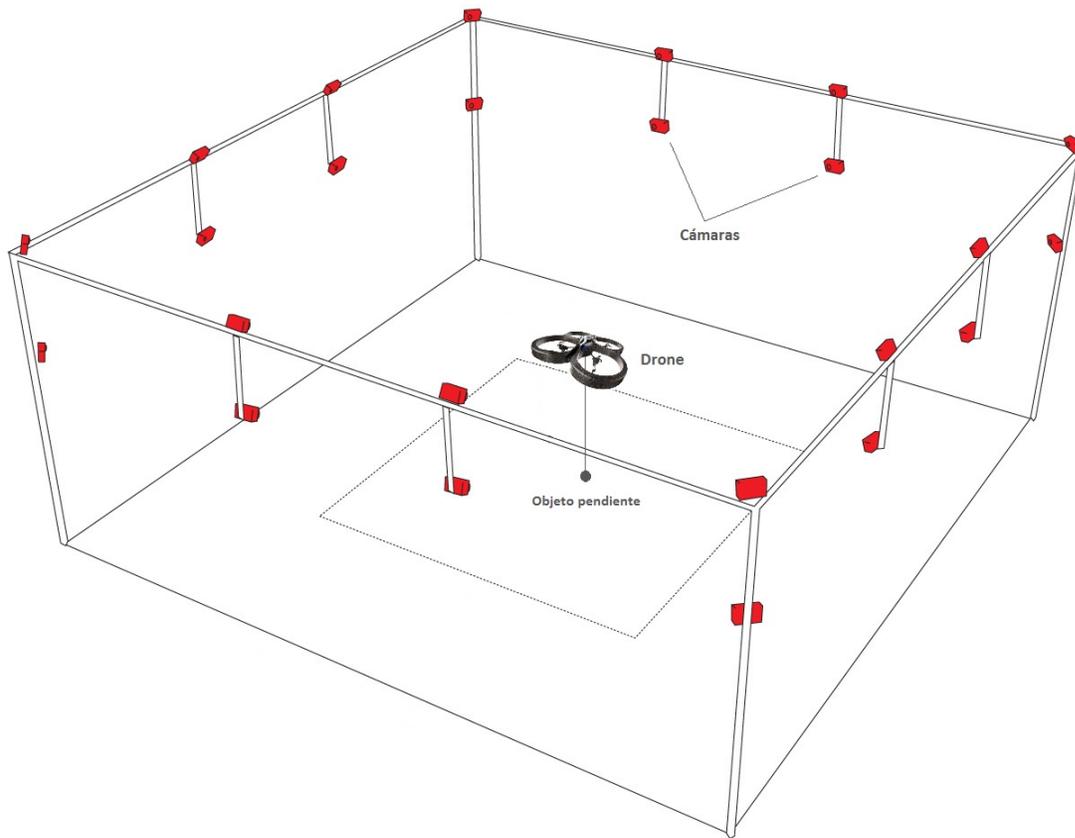
Como se mencionó, Parrot es el fabricante del AR-Drone. Además, distribuye un SDK para crear aplicaciones para los sistemas operativos iOS, Android, Linux y Windows para controlar el AR-Drone. Sin embargo, éste no es el único camino para lograrlo. Por ello y por simplicidad se optó por una ruta alternativa: utilizar el sistema ROS (Robot Operating System) para controlar el AR-Drone.

ROS es un sistema de código abierto que provee librerías y herramientas para ayudar a los desarrolladores de software a crear aplicaciones para robots. ROS provee abstracción del hardware, controladores de dispositivos, librerías, herramientas de visualización, comunicación por mensajes, administración de paquetes y más. ROS está bajo la licencia open source, BSD.

Este sistema es fácil de implementar en cualquier lenguaje de programación moderno como Python, C++, Lisp, etc. En la actualidad ROS se prueba principalmente en sistemas Ubuntu y Mac OS X, aunque la comunidad ROS ha proporcionado soporte para Fedora, Gentoo, ARch Linux y otras plataformas Linux.

El objetivo principal de ROS es apoyar la reutilización de código en la investigación y desarrollo de la robótica. ROS es un marco distribuido de procesos que permite a los ejecutables ser diseñados de forma individual y libremente acoplados en tiempo de ejecución. Estos procesos pueden agruparse en paquetes que pueden ser fácilmente compartidos y distribuidos.

Los principales elementos que intervienen en un proceso de ejecución de ROS son descritos a continuación:



**Figura 2.4.** Cámaras distribuidas en el espacio de trabajo

- **Nodos:** Un nodo es un ejecutable que usa ROS para comunicarse con otros nodos. Están diseñados para operar pequeños subprocesos. El sistema de control de un robot normalmente comprenderá muchos nodos; por ejemplo, un nodo controla un telémetro láser, un nodo controla la velocidad de las llantas del robot, un nodo lleva a cabo la planificación de ruta, etc.

El uso de nodos en ROS proporciona algunos beneficios para el sistema en general. Hay cierta tolerancia a fallas puesto que los accidentes se concentran en nodos individuales. Además, la complejidad del código se reduce en comparación a los sistemas monolíticos.

Todos los nodos que se ejecutan tienen un nombre de recurso gráfico que los identifica de forma única en el sistema.

- **Topics:** Son buses a través de los cuales los nodos intercambian información. Los nodos pueden publicar mensajes a un topic, o bien, suscribirse a él para recibir mensajes. Los topics no tienen información sobre quién publica o se suscribe a ellos. En general, los nodos no son concientes de con quién se comunican, sino

que sólo se interesan en publicar o suscribirse al topic adecuado. Puede haber múltiples nodos con publicaciones o suscripciones al mismo nodo.

Al igual que los nodos, los topics reciben un nombre único para ser identificados en el sistema.

- *Messages*: Los nodos se comunican entre sí mediante la publicación de messages a los topics. Un message es una estructura de datos simple. Se admiten tipos estándar (enteros, punto flotante, booleanos, etc.), así como los arreglos de tipos estándar. Los messages pueden incluir estructuras anidadas arbitrariamente y matrices.

Una versión de ROS puede ser incompatible con otra. Normalmente están referidas por un sobrenombre en vez de por una versión numérica. En esta tesis se optó por utilizar la versión *ROS Fuerte* (cuya guía de instalación en Ubuntu 12.04 se encuentra en [10]) aunque su elección puede ser arbitraria ya que el controlador para el AR-Drone es compatible para cualquier versión.

### 2.3.1. Controlador del AR-Drone

Para manipular el vuelo del AR-Drone es necesaria la instalación de un controlador, mismo que permita al usuario tener acceso a todos los recursos del drone en tiempo de vuelo.

Una guía de instalación del controlador para el AR-Drone se encuentra en [11].

La instalación de dicho controlador incluye un archivo ejecutable adicional denominado *keyboard controller*, el cual permite al usuario tener un control sobre el movimiento del cuadrirotor desde su teclado. Por default, las teclas de instrucción son:

- E - Cabeceo positivo, desplazamiento hacia adelante
- D - Cabeceo negativo, desplazamiento hacia atrás
- S - Alabeo negativo, desplazamiento hacia la izquierda
- F - Alabeo positivo, desplazamiento hacia la derecha
- W - Guiñada en sentido antihorario
- R - Guiñada en sentido horario
- Q - Desplazamiento vertical hacia arriba
- A - Desplazamiento vertical hacia abajo
- Y - Despegar y estabilizarse sobre un punto en el aire

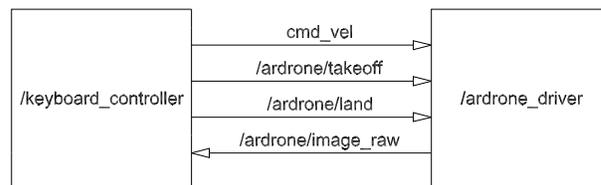
- H - Aterrizar
- SPACEBAR - Parada de emergencia

Los parámetros de configuración (máximos ángulos de alabeo y cabeceo y máximas velocidades de desplazamiento vertical y de guiñada) son establecidas dentro del archivo `/home/<user>/ros_workspace/ardrone_tutorials/launch/keyboard_controller.launch` que fue creado durante la instalación del controlador para el AR-Drone.

Una vez configurados dichos parámetros es posible ahora ejecutar el *keyboard controller* para manipular el vuelo del dron. Para ello, debe ejecutarse el siguiente comando desde una terminal en Ubuntu:

- `roslaunch ardrone_tutorials keyboard_controller.launch`

Mediante el comando anterior, ROS ejecuta los nodos y topics necesarios para establecer la comunicación con el dron desde la computadora. Un diagrama de los nodos y topics activos en esta operación es el mostrado en la Fig. 2.5. Los bloques de la figura representan los nodos y las flechas representan los topics a través de los cuales se transmiten los messages desde el publicador hacia el suscriptor.



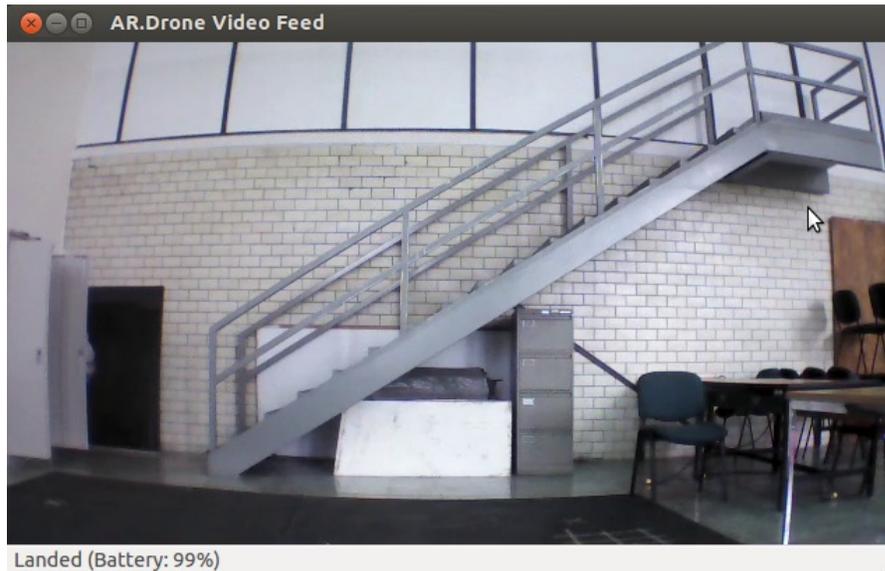
**Figura 2.5.** Conexión de los nodos *keyboard controller* y controlador del AR-Drone

El nodo */ardrone\_driver* es el controlador del AR-Drone. El nodo */keyboard\_controller* es la interface de comunicación entre el usuario y el dron, y despliega una ventana gráfica como la mostrada en la Fig. 2.6; a través de la cual se transmite en tiempo real el video obtenido por la cámara frontal del dron. Al mantener activa dicha ventana, el usuario puede comenzar a manipular el vuelo del dron con las teclas descritas.

Sin embargo, operar el dron de esa manera no es parte del objetivo de esta tesis.

### 2.3.2. Estrategia de comunicación

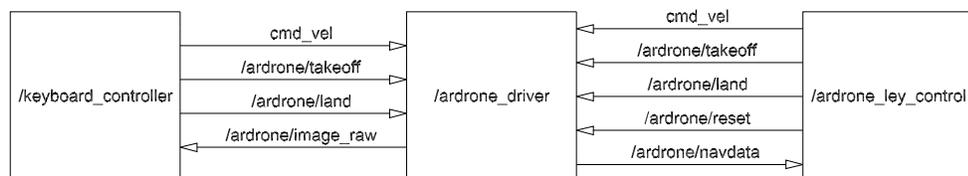
Para operar el dron se creará un nodo adicional en el cual se ejecuta una ley de control. Dicho nodo publicará en el topic correspondiente las señales de control derivadas de esa ley para lograr el control del objeto pendiente del cuadrirotor.



**Figura 2.6.** Ventana gráfica del *keyboard controller*

En [12] se describe el procedimiento para la creación de nodos y topics.

La Fig. 2.7 muestra un nuevo diagrama de comunicación entre nodos que incluye al nuevo nodo de control (llamado */ardrone\_ley\_control*) y los topics requeridos para el envío de información.



**Figura 2.7.** Conexión de los nodos *keyboard controller*, controlador del AR-Drone y nodo de control

A pesar de que el nodo */keyboard\_controller* no tiene aplicación directa en nuestro desarrollo, es conveniente mantenerlo activo debido a que proporciona la función de paro de emergencia en caso de un accidente no previsto.

ROS cuenta con el paquete *geometry\_msgs.msg* que incluye una estructura de datos llamada *Twist*. La estructura *Twist* está formada por seis elementos: *linear.x*, *linear.y*, *linear.z*, *angular.x*, *angular.y* y *angular.z*; todos de tipo *float64*. Al publicar un objeto de tipo *Twist* en el topic *cmd\_vel*, el controlador del driver atenderá a las

siguientes interpretaciones:

- objeto.linear.x: Ángulo de alabeo deseado
- objeto.linear.y: Ángulo de cabeceo deseado
- objeto.linear.z: Velocidad de desplazamiento vertical deseada
- objeto.angular.x: no implementado
- objeto.angular.y: no implementado
- objeto.angular.z: velocidad de guiñada deseada

Recuérdese que esas variables de control toman un valor dentro del rango  $[-1,1]$ , que representa una escala entre el valor máximo alcanzable en un sentido y el mismo valor máximo en sentido contrario, según la variable a controlar.

Además, también se mencionó que el movimiento del drone está restringido al plano vertical y, por lo tanto, podemos controlar el sistema únicamente controlando la velocidad de ascenso/descenso y el ángulo de alabeo del drone; es decir, manteniendo los ángulos de cabeceo ( $\theta$ ) y guiñada ( $\psi$ ) en cero. Así, las únicas variables de interés son *linear.y* y *linear.z*, mientras que el resto de las variables permanecen en cero.

Asimismo, en algunos de los topics utilizados en nuestro esquema de comunicación se requiere publicar mensajes sin contenido alguno (vacíos). Estos son mensajes de tipo *Empty* (paquete *std\_msgs.msg* de ROS). Cuando el nodo */ardrone\_driver* encuentra un mensaje *Empty* en el topic */ardrone/takeoff*, */ardrone/land* o */ardrone/reset*, realiza las operaciones de despegue, aterrizaje o reinicio del sistema del drone, respectivamente.

Por otra parte, a través del topic */ardrone/navdata* el AR-Drone proporciona todos los datos de navegación obtenidos mediante sus propios sensores a bordo; por ejemplo, ángulos y velocidades de orientación (alabeo, cabeceo y guiñada), velocidades cartesianas de desplazamiento, altitud respecto al suelo, nivel de batería, temperatura interna, etc. Sin embargo, dicha información presenta una variación significativa debida a la presencia de ruido en la medición de los sensores. Por lo tanto, la información de posición y orientación del drone será tomada desde el sistema OptiTrack por tener mayor precisión. Además, las velocidades de desplazamiento serán calculadas matemáticamente como se describirá en capítulos posteriores.

# Capítulo 3

## Modelado y control del sistema

### 3.1. Cuadrirotor de configuración “tipo X”

El AR-Drone de Parrot es un cuadrirotor de configuración “tipo X”, considerando que posee dos motores delanteros y dos motores traseros con distribución en forma de equis; a diferencia de otras configuraciones, por ejemplo, aquella que contempla un motor delantero y uno trasero, uno derecho y uno izquierdo.

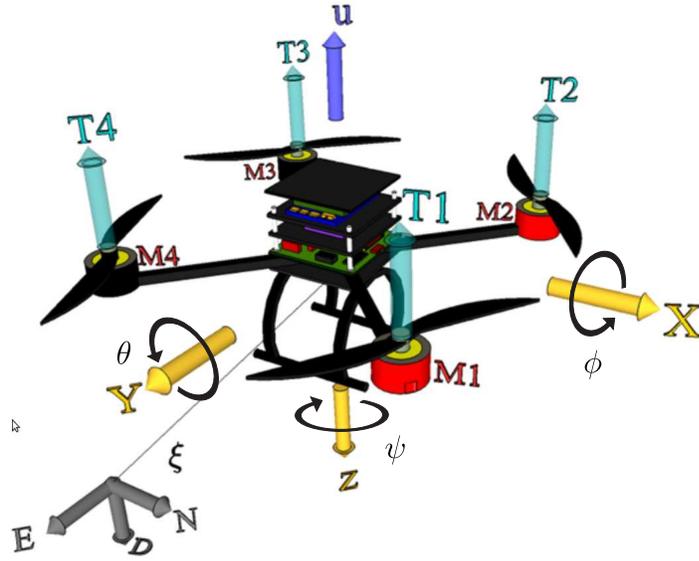
En [1] se encuentra el siguiente desarrollo basado en el formalismo Newton-Euler para la obtención del modelo dinámico general de un cuadrirotor en configuración “tipo X”.

Considérese un cuadrirotor representado en un marco de referencia inercial fijo del tipo Norte-Este-Abajo (*NED* por sus siglas en inglés) y en el marco inercial del cuadrirotor, como se aprecia en la Fig. 3.1. Sea  $\{N, E, D\}$  la representación del marco de referencia inercial fijo y  $\{X, Y, Z\}$  el marco de referencia del cuadrirotor. El vector de posición del centro de masa del cuadrirotor se denota por  $\xi = (x, y, z)^T$ , que representa las coordenadas de la aeronave respecto al marco de referencia NED. El vector de orientación del cuadrirotor con respecto al marco inercial es denotado por  $\eta = (\psi, \theta, \phi)^T$ , donde  $\psi$ ,  $\theta$  y  $\phi$  son los ángulos de guiñada, cabeceo y alabeo, respectivamente. La dinámica no lineal completa del cuadrirotor puede ser expresada mediante:

$$m\ddot{\xi} = -mgD + RF \quad (3.1)$$

$$I\dot{\Omega} = -\Omega \times I\Omega + \tau \quad (3.2)$$

donde  $D$  es un vector unitario sobre el eje  $\mathbf{D}$ ,  $R \in SO(3)$  es una matriz de rotación que asocia el marco inercial con el marco de referencia del cuadrirotor,  $F$  denota la fuerza de empuje total producida por la aeronave,  $m$  es la masa total,  $g$  es la constante de aceleración gravitacional,  $\Omega$  representa la velocidad angular del vehículo expresado en el marco de referencia del cuadrirotor,  $\tau$  es el vector de torques ge-



**Figura 3.1.** Diagrama NED del modelo dinámico del cuadricóptero. Imagen obtenida de [1].

neralizados que producen las orientaciones de guiñada, cabeceo y alabeo, expresado como  $\tau = \{\tau_\psi, \tau_\theta, \tau_\phi\}^T$  e  $I$  es la matriz de inercias de la forma

$$I = \begin{bmatrix} I_{xx} & 0 & 0 \\ 0 & I_{yy} & 0 \\ 0 & 0 & I_{zz} \end{bmatrix}$$

Sea  $u = \sum_{i=1}^4 T_i$  la fuerza de empuje producida por el cuadricóptero, donde  $T_i$  es fuerza de empuje producida por el motor  $M_i$ . Asumiendo que esa fuerza sólo tiene componente sobre el eje  $\mathbf{Z}$ , entonces la fuerza total de empuje es  $F = \{0, 0, -u\}^T$ . La matriz de rotación  $R$  es

$$R = \begin{bmatrix} c_\theta c_\psi & s_\phi s_\theta c_\psi - c_\phi s_\psi & c_\phi s_\theta c_\psi + s_\phi s_\psi \\ c_\theta s_\psi & s_\phi s_\theta s_\psi + c_\phi c_\psi & c_\phi s_\theta s_\psi - s_\phi c_\psi \\ -s_\theta & s_\phi c_\theta & c_\phi c_\theta \end{bmatrix} \quad (3.3)$$

donde  $c_* = \cos(*)$  y  $s_* = \sin(*)$ . Defínase ahora un vector auxiliar  $\tilde{\tau}$  relacionado al torque generalizado  $\tau$  y basándose en (3.2)

$$\tilde{\tau} = \begin{bmatrix} \tilde{\tau}_\psi \\ \tilde{\tau}_\theta \\ \tilde{\tau}_\phi \end{bmatrix} = I^{-1}W^{-1} \left( -I\dot{W}\dot{\eta} - W\dot{\eta} \times IW\dot{\eta} + \tau \right) \quad (3.4)$$

donde  $\Omega = W\dot{\eta}$  y  $W$  es

$$W = \begin{bmatrix} -\sin \theta & 0 & 1 \\ \cos \theta \sin \phi & \cos \phi & 0 \\ \cos \theta \cos \phi & -\sin \phi & 0 \end{bmatrix}$$

A partir de (3.1)-(3.4), es posible obtener un modelo dinámico para el cuadrirotor

$$\begin{aligned} m\ddot{x} &= -u (\cos \psi \sin \theta \cos \phi + \sin \psi \sin \phi) \\ m\ddot{y} &= -u (\sin \psi \sin \theta \cos \phi - \cos \psi \sin \phi) \\ m\ddot{z} &= -u (\cos \theta \cos \phi) + mg \\ \ddot{\psi} &= \tilde{\tau}_\psi \\ \ddot{\theta} &= \tilde{\tau}_\theta \\ \ddot{\phi} &= \tilde{\tau}_\phi \end{aligned} \quad (3.5)$$

En el cuadrirotor en configuración “tipo X”, los motores  $M_1$  y  $M_3$  giran en sentido horario, mientras que los motores  $M_2$  y  $M_4$  giran en sentido antihorario. Asumiendo que el empuje total contrarresta el efecto de la fuerza gravitacional, i.e., el cuadrirotor se encuentra en *hover*<sup>1</sup>, podemos considerar que cada empuje puede ser modelado de la forma  $T_i = Cw_i^2$ , donde  $C$  es una constante que depende de las características del motor y  $w_i$  es la velocidad del motor  $i$ . Por simplicidad también se asume que el torque  $\tau_i$  generado por cada motor es proporcional a su fuerza de sustentación, entonces  $\tau_i = C_M f_i$ . Tomando en cuenta las suposiciones previas, podemos obtener los torques generalizados mediante

$$\begin{bmatrix} \tau_\psi \\ \tau_\theta \\ \tau_\phi \end{bmatrix} = \begin{bmatrix} -C_M & C_M & -C_M & C_M \\ -l & -l & l & l \\ -l & l & l & -l \end{bmatrix} \begin{bmatrix} T_1 \\ T_2 \\ T_3 \\ T_4 \end{bmatrix}$$

---

<sup>1</sup>Condición que se produce cuando el cuadrirotor puede mantenerse en equilibrio respecto al eje vertical

donde  $l$  es la distancia entre el centro de masas del cuadrirotor y el centro del motor.

## 3.2. Caracterización del AR-Drone

Para realizar maniobras en el espacio con el AR-Drone es necesario controlar la velocidad a la que giran sus cuatro motores. Con estos cambios se producen variaciones en los ángulos de alabeo  $\phi$ , cabeceo  $\theta$  y guiñada  $\psi$ , logrando además desplazamientos sobre el sistema de coordenadas  $XYZ$  (Fig. 3.4).

El AR-Drone ya posee un control interno de su fabricante que se encarga de regular las velocidades a las que cada hélice debe girar para lograr las maniobras deseadas [2]. Dicho control de navegación permite la entrada de cuatro variables de control:

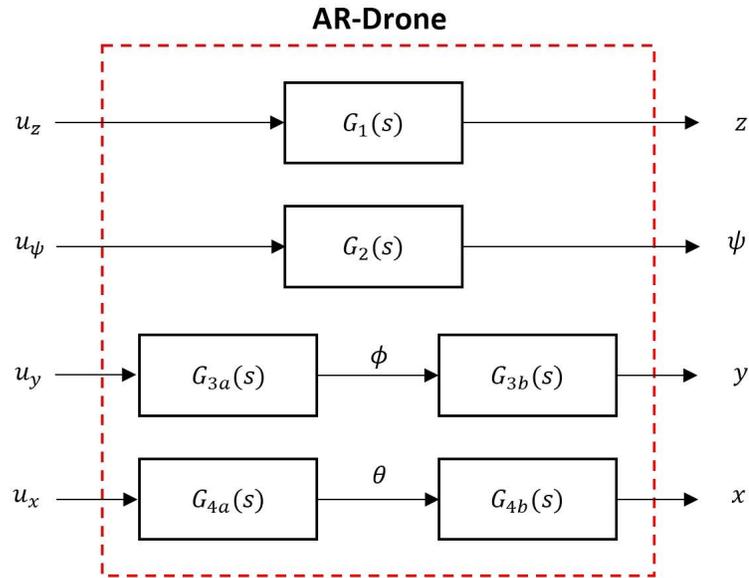
- La velocidad deseada de ascenso/descenso ( $u_z$ )
- El ángulo deseado de alabeo ( $u_y$ )
- El ángulo deseado de cabeceo ( $u_x$ )
- La velocidad deseada de guiñada ( $u_\psi$ )

Debido a la naturaleza hermética del control interno, el usuario no tiene acceso a los estados del sistema ni control directo sobre las velocidades de cada motor. En lugar de ello, el AR-Drone permite al usuario definir previamente (desde la programación en ROS) los ángulos máximos alcanzables de alabeo y cabeceo, así como las velocidades máximas de desplazamiento vertical y de guiñada. Luego, por cada uno de esos argumentos el drone espera recibir como instrucción un valor numérico decimal entre 1 y  $-1$ , siendo éste el rango de escalamiento entre el valor máximo alcanzable predefinido en un sentido y el mismo valor máximo en el sentido opuesto. Dicha instrucción es interpretada por el control del drone como un ángulo de alabeo o cabeceo, o bien, como una velocidad de desplazamiento vertical o de guiñada que debe ser alcanzada por el propio drone, según sea el caso.

Por lo anterior, en lugar de modelar el AR-Drone como un cuadrirotor estándar, i.e., considerando las velocidades de giro de sus hélices como variables de control y sus ángulos de orientación como salidas, debemos modelar el drone incluyendo su propio control interno. Puesto que el control interno es capaz de alcanzar y mantener los ángulos de orientación deseados y la velocidad vertical deseada, no es necesario lidiar con la complejidad del modelo matemático en (3.5). En lugar de ello, es posible considerar al drone como un sistema cuyas variables de entrada son los valores deseados de alabeo, cabeceo, velocidad de guiñada y velocidad vertical; y como estados del sistema los valores reales de los ángulos de orientación, la posición y velocidad del drone [3].

Por lo anterior se propone modelar al AR-Drone como un conjunto de sistemas lineales tipo “caja negra” (Fig. 3.2), en la que se desconoce la dinámica entre las

velocidades de rotación de las hélices y las posiciones, velocidades o aceleraciones del cuerpo.



**Figura 3.2.** Aproximación del AR-Drone por “caja negra”

De [3] se sabe que  $G_1(s)$  y  $G_2(s)$  en la Fig. 3.2 son funciones de transferencia de segundo orden de la forma

$$\begin{aligned} G_1(s) &= \frac{Z(s)}{U_z(s)} = \frac{k_z}{s^2 + p_z s} \\ G_2(s) &= \frac{\Psi(s)}{U_\psi(s)} = \frac{k_\psi}{s^2 + p_\psi s} \end{aligned} \quad (3.6)$$

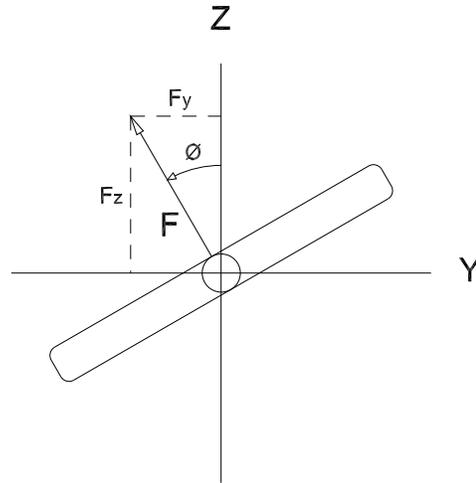
para  $-1 \leq u_z, u_\psi \leq 1$ ; mientras que de [2] se sabe que  $G_{3a}(s)$  y  $G_{4a}(s)$  son funciones de transferencia de segundo orden de la forma

$$\begin{aligned} G_{3a}(s) &= \frac{\Phi(s)}{U_y(s)} = -\frac{k_\phi \omega_\phi^2}{s^2 + 2\xi_\phi \omega_\phi s + \omega_\phi^2} \\ G_{4a}(s) &= \frac{\Theta(s)}{U_x(s)} = \frac{k_\theta \omega_\theta^2}{s^2 + 2\xi_\theta \omega_\theta s + \omega_\theta^2} \end{aligned} \quad (3.7)$$

para  $-1 \leq u_y, u_x \leq 1$ .

A continuación se pretende determinar el orden de las funciones  $G_{3b}(s)$  y  $G_{4b}(s)$ . Del diagrama de cuerpo libre mostrado en la Fig. 3.3 se obtiene

$$F_z = F \cos \phi = mg \quad (3.8)$$



**Figura 3.3.** Fuerza de sustentación del drone

donde  $F$  es la fuerza de sustentación del drone,  $F_z$  su componente vertical,  $F_y$  su componente horizontal,  $m$  la masa del drone y  $g$  la constante de aceleración gravitacional. Para pequeños ángulos de alabeo y cabeceo se puede realizar la aproximación

$$\sin \phi \approx \phi \quad y \quad \cos \phi \approx 1 \quad (3.9)$$

entonces, de (3.8) se tiene

$$F \approx mg \quad (3.10)$$

luego, del mismo diagrama de la Fig. 3.3 se tiene

$$m\ddot{y} = -F \sin \phi \quad (3.11)$$

y utilizando la aproximación dada en (3.9), de la ecuación (3.11) se llega a

$$m\ddot{y} = -F\phi \quad (3.12)$$

Sustituyendo (3.10) en (3.12) y reduciendo se obtiene

$$\ddot{y} = -g\phi \quad (3.13)$$

Al aplicar la transformada de Laplace a (3.13) resulta

$$s^2Y(s) = -g\Phi(s) \quad (3.14)$$

luego, reagrupando términos de (3.14) encontramos la primera función de transferencia buscada

$$G_{3b}(s) = \frac{Y(s)}{\Phi(s)} = -\frac{g}{s^2} \quad (3.15)$$

De manera análoga, siguiendo el mismo procedimiento, se obtiene la segunda función de transferencia

$$G_{4b}(s) = \frac{X(s)}{\Theta(s)} = \frac{g}{s^2} \quad (3.16)$$

Finalmente, a partir de las ecuaciones (3.6), (3.7), (3.15) y (3.16) es posible determinar la forma y orden de las cuatro funciones de transferencia para el modelado tipo “caja negra” de la Fig. 3.2:

$$G_1(s) = \frac{Z(s)}{U_z(s)} = \frac{k_z}{s^2 + p_z s} \quad (3.17)$$

$$G_2(s) = \frac{\Psi(s)}{U_\psi(s)} = \frac{k_\psi}{s^2 + p_\psi s} \quad (3.18)$$

$$G_3(s) = \frac{Y(s)}{U_y(s)} = \frac{gk_\phi\omega_\phi^2}{s^4 + 2\xi_\phi\omega_\phi s^3 + \omega_\phi^2 s^2} \quad (3.19)$$

$$G_4(s) = \frac{X(s)}{U_x(s)} = \frac{gk_\theta\omega_\theta^2}{s^4 + 2\xi_\theta\omega_\theta s^3 + \omega_\theta^2 s^2} \quad (3.20)$$

para  $-1 \leq u_z, u_\psi, u_y, u_x \leq 1$ . Desafortunadamente, resulta una tarea complicada determinar con exactitud los valores de  $k$ ,  $p$ ,  $\xi$  y  $\omega$  en (3.17)-(3.20) debido a la presencia de perturbaciones internas y externas en el dron (por ejemplo, calentamiento de las baterías, nivel de carga de las baterías, viento, etc).

En [4], [5] y [6] se muestran distintas estrategias experimentales que se emplearon para estimar el valor de dichos parámetros; sin embargo, los resultados obtenidos son diferentes entre sí.

### 3.3. Sistema cuadrirotor-péndulo

Considérese al AR-Dron como una grúa viajera que puede desplazarse sobre un plano vertical y de la cual se encuentra suspendido un objeto a través de una cuerda.

Para modelar este sistema se consideran las siguientes suposiciones:

- El momento de inercia del objeto pendiente es despreciable.
- El objeto pendiente puede ser considerado como una masa puntual.
- La cuerda es considerada rígida y de masa despreciable.
- El movimiento de la grúa se restringe al plano vertical.

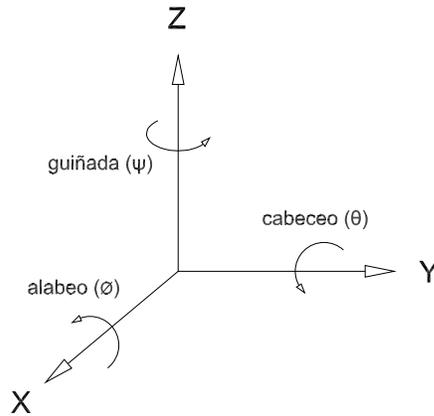
El modelado del sistema cuadrirotor-péndulo se realizará dentro del marco de referencia inercial fijo con la orientación mostrada en la Fig. 3.4, donde además, se muestran los ángulos de alabeo  $\phi$ , cabeceo  $\theta$  y guiñada  $\psi$  del dron.

Sea  $YZ$  el plano vertical al cual se restringe el movimiento del sistema cuadrirotor-péndulo y se asume que el sistema no abandona dicho plano. Puesto que el movimiento del dron se encuentra restringido al plano  $YZ$ , podemos controlar el sistema únicamente controlando la velocidad de ascenso/descenso y el ángulo de alabeo del dron; es decir, manteniendo los ángulos de cabeceo ( $\theta$ ) y guiñada ( $\psi$ ) en cero.

De la Fig. 3.4 se observa que un cambio en el ángulo de alabeo  $\phi$  producirá un desplazamiento sobre el eje  $Y$ , mientras que un cambio en la velocidad de desplazamiento vertical producirá un desplazamiento sobre el eje  $Z$ .

Sean  $u_y$  y  $u_z$  las variables de control que el dron espera recibir como argumento para controlar su ángulo de alabeo  $\phi$  y su velocidad de desplazamiento vertical  $\dot{z}$ , respectivamente. Ambas variables de control toman valores dentro del rango  $[-1, 1]$ , que representa una escala entre el valor máximo alcanzable en un sentido y el mismo valor máximo en sentido contrario, según la variable a controlar.

La dinámica entre las variables de entrada  $u_z$  y  $u_y$  y las salidas  $z$  y  $y$  del cuadrirotor está determinada por las funciones de transferencia (3.17) y (3.19), respectivamente. De dichas funciones se obtienen las ecuaciones diferenciales mostradas en (3.21).



**Figura 3.4.** Ángulos de orientación en el cuadrirotor

$$\begin{aligned} y^{(4)} + a_2 y^{(3)} + a_3 \ddot{y} &= a_1 u_y \\ \ddot{z} + b_2 \dot{z} &= b_1 u_z \end{aligned} \quad (3.21)$$

con

$$a_1 = gk_\phi \omega_\phi^2 \quad a_2 = 2\xi_\phi \omega_\phi \quad a_3 = \omega_\phi^2 \quad b_1 = k_z \quad b_2 = p_z$$

Ahora, es necesario modelar la dinámica del péndulo para obtener un modelo matemático del sistema cuadrirotor-péndulo.

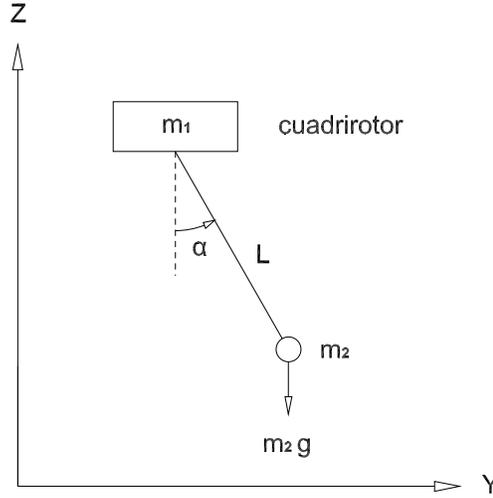
La Fig. 3.5 muestra un diagrama esquemático de la grúa viajera sobre el plano  $YZ$  con una masa pendiente, donde  $m_1$  es la masa del cuadrirotor,  $m_2$  la masa del objeto pendiente y  $L$  la longitud de la cuerda.

La posición de  $m_2$  en el plano vertical está dada por las coordenadas

$$\begin{aligned} y_{m_2} &= y_{m_1} + L \sin \alpha \\ z_{m_2} &= z_{m_1} - L \cos \alpha \end{aligned}$$

cuyas segundas derivadas con respecto al tiempo son

$$\begin{aligned} \ddot{y}_{m_2} &= \ddot{y}_{m_1} - L\dot{\alpha}^2 \sin \alpha + L\ddot{\alpha} \cos \alpha \\ \ddot{z}_{m_2} &= \ddot{z}_{m_1} + L\dot{\alpha}^2 \cos \alpha + L\ddot{\alpha} \sin \alpha \end{aligned} \quad (3.22)$$



**Figura 3.5.** Diagrama del sistema cuadrirotor-péndulo

De acuerdo a la segunda ley de Newton

$$\begin{aligned} m_2 \ddot{y}_{m_2} &= -T \sin \alpha \\ m_2 \ddot{z}_{m_2} &= T \cos \alpha - m_2 g \end{aligned} \quad (3.23)$$

donde  $T$  es la tensión de la cuerda y  $g$  es la constante de aceleración gravitacional. Sustituyendo (3.22) en (3.23) y resolviendo el sistema de ecuaciones para  $\ddot{\alpha}$  se obtiene

$$\ddot{\alpha} = -\frac{\ddot{y}_{m_1} \cos \alpha + \ddot{z}_{m_1} \sin \alpha + g \sin \alpha}{L} \quad (3.24)$$

La ecuación diferencial (3.24) representa el modelo dinámico del péndulo donde las aceleraciones cartesianas  $\ddot{y}_{m_1}$  y  $\ddot{z}_{m_1}$  del drone se consideran las variables de entrada para el péndulo y  $\alpha$ ,  $\dot{\alpha}$  son nuevas variables de estados.

Nótese de la ecuación (3.24) que el comportamiento del péndulo es afectado directamente por el movimiento del drone.

Debido a que la masa del objeto pendiente es pequeña (14 g) en comparación con la masa del drone (460 g), se asume que ésta no provoca perturbaciones significativas en el cuadrirotor; es decir, el comportamiento del cuadrirotor no es afectado por las oscilaciones del péndulo. Por lo tanto, para el modelado de este sistema se considera que el comportamiento del péndulo está en función del comportamiento del drone, como lo muestra la ecuación (3.24), pero el comportamiento del drone no

será afectado por el movimiento péndulo tal como se observa en (3.21).

### Representación del sistema en variables de estado

Las ecuaciones diferenciales dadas en (3.21) y (3.24) describen un modelo matemático completo para el sistema. Considérese ahora el siguiente cambio de variables:

$$\begin{aligned} x_1 &= y_{m_1}, & x_2 &= \dot{y}_{m_1}, & x_3 &= \ddot{y}_{m_1}, & x_4 &= y_{m_1}^{(3)} \\ x_5 &= z_{m_1}, & x_6 &= \dot{z}_{m_1}, & x_7 &= \alpha, & x_8 &= \dot{\alpha} \end{aligned} \quad (3.25)$$

En estas coordenadas, el sistema cuadricóptero-péndulo queda descrito por las siguientes ecuaciones:

$$\dot{x} = \begin{bmatrix} x_2 \\ x_3 \\ x_4 \\ -a_2 x_4 - a_3 x_3 \\ x_6 \\ -b_2 x_6 \\ x_8 \\ -\frac{x_3 \cos x_7 - (b_2 x_6 - g) \sin x_7}{L} \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ a_1 & 0 \\ 0 & 0 \\ 0 & b_1 \\ 0 & 0 \\ 0 & -\frac{b_1 \sin x_7}{L} \end{bmatrix} \begin{bmatrix} u_y \\ u_z \end{bmatrix} \quad (3.26)$$

Puesto que el objetivo de control es que la carga siga una trayectoria deseada sobre el plano  $YZ$ , la salida del sistema es la posición de la carga en ese plano. Así, se definen entonces las funciones de salida del sistema:

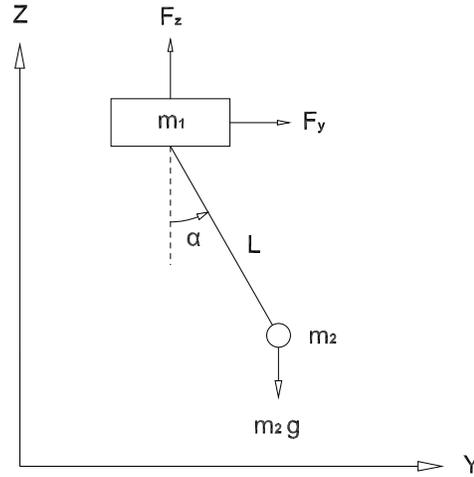
$$\begin{aligned} h_1(x) &= x_1 + L \sin x_7 \\ h_2(x) &= x_5 - L \cos x_7 \end{aligned} \quad (3.27)$$

Se propone abordar el problema del control del sistema a través de una retroalimentación de estados. La principal desventaja de este tipo de control es que se requiere del conocimiento del modelo exacto del sistema, lo cual resulta inconveniente para nuestro caso puesto que, como se mencionó antes, no es posible determinar con exactitud los valores de  $a_i$  y  $b_i$  en (3.26).

Por lo anterior, se propone entonces la siguiente estrategia de control.

Provisionalmente, considérese un sistema similar al anterior, con la única diferencia de que el cuadrirotor es ahora una masa inerte que puede desplazarse sobre el plano  $YZ$  impulsada por dos fuerzas  $F_y$  y  $F_z$ , tal como se observa en la Fig. 3.6.

Considérese nuevamente que el movimiento de la masa  $m_1$  no es afectado por las



**Figura 3.6.** Sistema provisional, masas unidas por una cuerda rígida

oscilaciones del péndulo. De este modo, la ecuación (3.24) es válida también para este sistema provisional. De acuerdo a la segunda ley de Newton, la dinámica de la masa  $m_1$  es

$$\begin{aligned} m_1 \ddot{y}_{m_1} &= F_y \\ m_1 \ddot{z}_{m_1} &= F_z - m_1 g \end{aligned} \quad (3.28)$$

Siguiendo el mismo cambio de variables que en (3.25), a partir de (3.24) y (3.28) se obtiene el modelo matemático para el sistema provisional

$$\dot{x} = f(x) + g_1(x)F_y + g_2(x)F_z \quad (3.29)$$

donde

$$f(x) = \begin{bmatrix} x_2 \\ 0 \\ 0 \\ 0 \\ x_6 \\ -g \\ x_8 \\ 0 \end{bmatrix} \quad g(x) = \begin{bmatrix} 0 & 0 \\ 1/m_1 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 1/m_1 \\ 0 & 0 \\ -\frac{\cos x_7}{m_1 L} & -\frac{\sin x_7}{m_1 L} \end{bmatrix}$$

La salida del sistema sigue siendo la establecida en (3.27).

A continuación se pretende diseñar un control por retroalimentación de estados que logre un desacoplamiento entrada-salida de este nuevo sistema. Para ello es necesario derivar con respecto al tiempo la salida del sistema hasta encontrar las entradas de control  $F_y$  y  $F_z$ . Realizando dicho procedimiento se puede encontrar la entrada de control en la derivada de segundo orden de la salida, a saber:

$$h^{(2)} = L_f^2 h(x) + L_g L_f h(x) F \quad (3.30)$$

con

$$L_f^2 h(x) = \begin{bmatrix} -Lx_8^2 \sin x_7 \\ Lx_8^2 \cos x_7 - g \end{bmatrix} \quad (3.31)$$

$$L_g L_f h(x) = \frac{1}{m_1} \begin{bmatrix} \sin^2 x_7 & -\sin x_7 \cos x_7 \\ -\sin x_7 \cos x_7 & \cos^2 x_7 \end{bmatrix} \quad (3.32)$$

Nótese que  $L_f h(x)$  es la derivada de  $h$  a lo largo del campo vectorial  $f$  (Derivada de Lie).

La matriz de desacoplamiento (3.32) tiene un rango igual a uno. En consecuencia, el grado relativo del sistema no está bien definido ya que se requiere que dicha matriz sea no singular. Por lo tanto, el desacoplamiento entrada-salida no puede ser conseguido por retroalimentación estática del estado.

A continuación se emplea el algoritmo de Extensión Dinámica [7].

Es necesario linealizar el comportamiento desde la entrada hacia una salida escalar del sistema. Sean  $w_1$  y  $w_2$  dos nuevas variables de control y considérese el siguiente cambio de variable:

$$F_y = w_1 \quad (3.33)$$

además, de (3.30) se tiene

$$h_2^{(2)} = L_f^2 h_2 + \frac{1}{m_1} \begin{bmatrix} -\sin x_7 \cos x_7 & \cos^2 x_7 \end{bmatrix} \begin{bmatrix} F_y \\ F_z \end{bmatrix} = w_2$$

entonces

$$F_z = \frac{m_1 w_2 - m_1 L_f^2 \ddot{h}_2 + w_1 \sin x_7 \cos x_7}{\cos^2 x_7} \quad (3.34)$$

Al sustituir (3.33) y (3.34) en (3.30) se obtiene

$$\begin{bmatrix} \ddot{h}_1 \\ \ddot{h}_2 \end{bmatrix} = \begin{bmatrix} -g \tan x_7 \\ 0 \end{bmatrix} + \begin{bmatrix} 0 & -\tan x_7 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} w_1 \\ w_2 \end{bmatrix} \quad (3.35)$$

Nótese que se ha optado por obtener  $F_z$  a partir de  $\ddot{h}_2$  para evitar una singularidad como ocurriría con  $\ddot{h}_1$  debido a la división por  $-\sin x_7 \cos x_7$ , el cual es cero en el punto de equilibrio del sistema.

Con esta linealización entrada-salida de  $h_2$ , se descubre en (3.35) que las derivadas de más alto orden de  $h_1$  y  $h_2$  que son afectadas por una entrada (en este caso  $\ddot{h}_1$  y  $\ddot{h}_2$ ) son afectadas por  $w_2$  y ninguna por  $w_1$ . Para conseguir un grado relativo bien definido, es necesario “retrasar” la aparición de  $w_2$  y esperar que aparezca al mismo tiempo que  $w_1$  en derivadas de mayor orden de  $h_1$  y  $h_2$ . La manera más simple de lograr esto es configurar  $w_2$  como la salida de un integrador cuya señal de entrada sea una nueva variable de control. Deberán agrarse tantos integradores como sea necesario a fin de lograr la aparición simultánea de las nuevas entradas de control.

Después de que dos integradores son colocados antes de la entrada  $w_2$  de la siguiente manera

$$w_2 = \xi_1 \quad \dot{\xi}_1 = \xi_2 \quad \dot{\xi}_2 = q_2 \quad w_1 = q_1 \quad (3.36)$$

con  $q_1$  y  $q_2$  las nuevas variables de control, se obtiene el siguiente sistema extendido:

$$\begin{aligned}
\dot{x}_1 &= x_2 \\
\dot{x}_2 &= q_1/m_1 \\
\dot{x}_3 &= 0 \\
\dot{x}_4 &= 0 \\
\dot{x}_5 &= x_6 \\
\dot{x}_6 &= \frac{\xi_1 + g - Lx_8^2 \cos x_7 + (q_1/m_1) \sin x_7 \cos x_7}{\cos^2 x_7} - g \\
\dot{x}_7 &= x_8 \\
\dot{x}_8 &= \frac{-q_1 \cos x_7 - \xi_1 m_1 \sin x_7 - m_1 g \sin x_7 + Lm_1 x_8^2 \sin x_7 \cos x_7}{Lm_1 \cos^2 x_7} \\
\dot{\xi}_1 &= \xi_2 \\
\dot{\xi}_2 &= q_2
\end{aligned}$$

Con los dos integradores agregados al sistema, ahora las nuevas variables de control  $q_1$  y  $q_2$  aparecen en la cuarta derivada de la salida con respecto al tiempo:

$$h^{(4)} = L_f^4 h(x) + L_g L_f^3 h(x) q$$

donde la matriz de desacoplamiento

$$L_g L_f^3 h(x) = \begin{bmatrix} \frac{g+\xi_1}{Lm_1 \cos^3 x_7} & -\tan x_7 \\ 0 & 1 \end{bmatrix} \quad (3.37)$$

es no singular para todos los valores de  $x_7$ , con excepción de  $x_7 = \pm \frac{n\pi}{2}$ , con  $n = 1, 2, 3, \dots$ , sin embargo, por razones físicas del sistema real esta condición no se produce, entonces esta matriz permanece no singular. Obsérvese además que otra singularidad existe cuando

$$\xi_1 = -g \quad (3.38)$$

además, de acuerdo a (3.35) y (3.36) se obtiene

$$\xi_1 = \ddot{h}_2 \quad (3.39)$$

Las ecuaciones (3.38) y (3.39) determinan que dicha singularidad en la matriz de desacoplamiento (3.37) existe cuando la segunda salida del sistema  $h_2$  presenta una aceleración negativa sobre el eje  $Z$  igual a la aceleración gravitacional.

Ahora, con los desarrollos anteriores es posible establecer la siguiente ley de control:

$$q = (L_g L_f^3 h(x))^{-1} (v - L_f^4 h(x)) \quad (3.40)$$

donde  $v$  es un control auxiliar que está dado por

$$v = h_d^{(4)} + k_1 (h_d^{(3)} - h^{(3)}) + k_2 (\ddot{h}_d - \ddot{h}) + k_3 (\dot{h}_d - \dot{h}) + k_4 (h_d - h)$$

siendo  $h_d(t)$  la trayectoria deseada para el objeto pendiente (parametrizada con respecto al tiempo) y los parámetros  $k_i$  son tales que el polinomio

$$s^4 + k_1 s^3 + k_2 s^2 + k_3 s + k_4$$

es Hurwitz<sup>2</sup>.

De acuerdo a la ley de control dada por (3.33), (3.34), (3.36) y (3.40) es posible conocer las fuerzas  $F_y$  y  $F_z$  requeridas para acelerar al dron y lograr el seguimiento de trayectoria deseada para la masa  $m_2$ . Ahora bien, considérense dichas fuerzas como fuerzas deseadas  $F_{y_d}$  y  $F_{z_d}$  y, de acuerdo a la segunda ley de Newton, estas fuerzas deseadas pueden convertirse en aceleraciones cartesianas deseadas en el dron:

$$\begin{aligned} \ddot{y}_d &= \frac{F_{y_d}}{m_1} \\ \ddot{z}_d &= \frac{F_{z_d}}{m_1} - g \end{aligned} \quad (3.41)$$

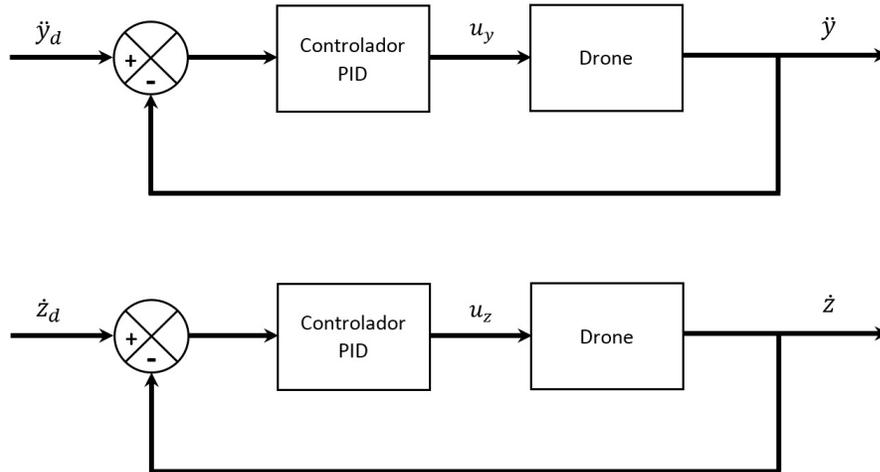
Ahora bien, los perfiles de aceleración  $\ddot{y}_d$  y  $\ddot{z}_d$  del dron dados en (3.41) garantizan que  $h_1$  y  $h_2$  sigan asintóticamente las trayectorias deseadas  $h_{1_d}$  y  $h_{2_d}$ . Luego, debe

---

<sup>2</sup>Criterio de Routh-Hurwitz. Este criterio es un método algebraico que proporciona información sobre la estabilidad absoluta de un sistema lineal e invariante con el tiempo que tiene una ecuación característica con coeficientes constantes. El criterio prueba si cualquiera de las raíces de la ecuación característica está en el semiplano izquierdo del plano complejo  $s$ . También indica el número de raíces que están sobre el eje imaginario  $j\omega$  y en el semiplano derecho.

diseñarse una segunda ley de control a fin de conseguir que las aceleraciones cartesianas  $\ddot{y}$  y  $\ddot{z}$  del drone, cuyo movimiento sobre el plano  $YZ$  está gobernado por las funciones de transferencia en (3.17) y (3.19), converjan a las aceleraciones deseadas  $\ddot{y}_d$  y  $\ddot{z}_d$  dadas en (3.41).

Nótese que dichas funciones de transferencia  $G_1(s)$  y  $G_3(s)$  están desacopladas entre sí. Por tal motivo se propone el empleo de dos controladores PID como se muestra en la Fig. 3.7. El empleo de un controlador PID tiene por ventaja su robustez y el hecho de que solo controla una variable de estado del sistema. Como se observa en dicha figura, en el primer lazo se controla la aceleración  $\ddot{y}$  del drone; sin embargo, en el segundo lazo no es posible controlar la aceleración  $\ddot{z}$  puesto que ésta no es una variable de estado del sistema, de acuerdo a la definición de estados dada en (3.25). Por lo tanto, el segundo lazo de control tomará como referencia la velocidad  $\dot{z}_d$ , misma que puede ser obtenida por integración de  $\ddot{z}_d$  en (3.41).



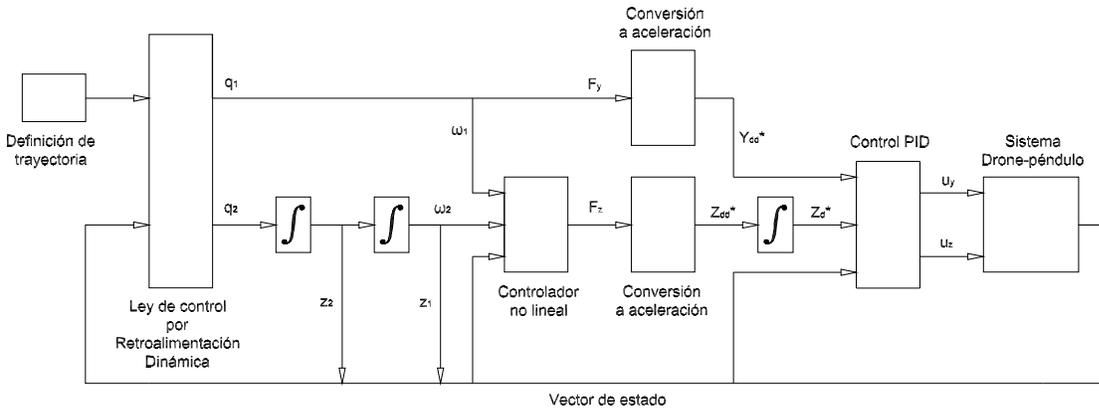
**Figura 3.7.** Controlador interno PID

Los controladores PID están definidos por las siguientes ecuaciones cuya salida son las variables de control  $u_y$  y  $u_z$ :

$$\begin{aligned} u_y &= k_{py} (\ddot{y}_d - \ddot{y}) + k_{dy} \frac{d}{dt} (\ddot{y}_d - \ddot{y}) + k_{iy} \int_0^t (\ddot{y}_d - \ddot{y}) dt \\ u_z &= k_{pz} (\dot{z}_d - \dot{z}) + k_{dz} \frac{d}{dt} (\dot{z}_d - \dot{z}) + k_{iz} \int_0^t (\dot{z}_d - \dot{z}) dt \end{aligned} \quad (3.42)$$

El esquema de control planteado en este capítulo se muestra en la figura 3.8. En la figura se aprecia un lazo de control compuesto por una retroalimentación dinámica

cuya salida es convertida a una nueva trayectoria deseada, misma que es alcanzada a través de un control PID.



**Figura 3.8.** Esquema de control por Retroalimentación Dinámica más PD

# Capítulo 4

## Estimación de estados del sistema

Los esquemas de control desarrollados en el capítulo anterior dependen de los ocho estados del sistema. Desafortunadamente, la medición de algunos de ellos, por ejemplo, la velocidad horizontal  $\dot{y}$  del dron ( $x_2$ ), su aceleración  $\ddot{y}$  ( $x_3$ ), su sobreaceleración  $y^{(3)}$  ( $x_4$ ), su velocidad vertical  $\dot{z}$  ( $x_6$ ) así como la velocidad angular  $\dot{\alpha}$  del péndulo ( $x_8$ ), representa una fuerte dificultad técnica. Una posible solución alternativa a la medición directa de estas variables es el empleo de un observador. Un observador es un objeto matemático (regularmente una ecuación diferencial) que construye una aproximación del vector de estado. La principal desventaja que presentan algunos observadores es que trabajan bajo la suposición de conocer el modelo exacto del sistema, lo cual resulta inconveniente para nuestro caso puesto que, como se mencionó antes, no es posible determinar con exactitud los valores de  $a_i$  y  $b_i$  en (3.26).

En este capítulo se describe un método sencillo y eficaz para estimar los estados no medibles del sistema.

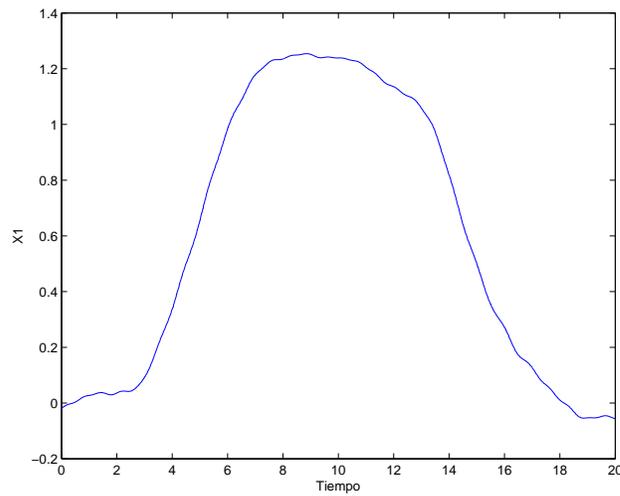
Nótese del modelo del sistema en (3.26) que algunos estados que pueden ser obtenidos mediante derivación con respecto al tiempo como se muestra en (4.1).

$$x_i(t) = \frac{dx_{i-1}(t)}{dt}, \quad i = 2, 3, 4, 6, 8 \quad (4.1)$$

Algunos estados como la posición  $y$  del dron ( $x_1$ ), su posición  $z$  ( $x_5$ ) y el ángulo de balanceo  $\alpha$  del péndulo ( $x_7$ ) no necesitarán estimarse puesto que están disponibles por medición directa.

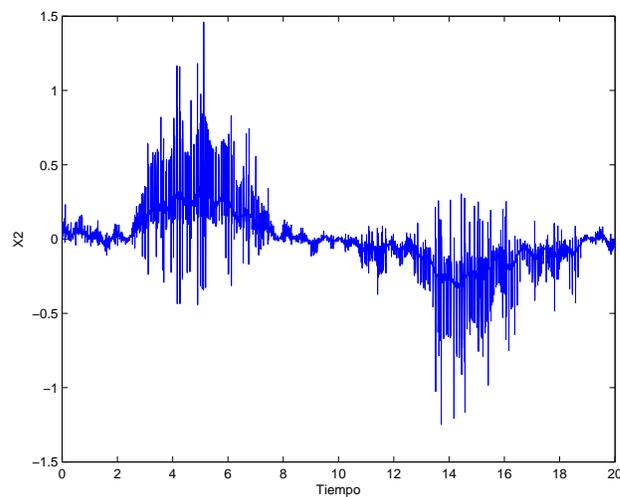
En la Fig. 4.1 se grafica el desplazamiento  $y$  del dron en un experimento realizado en tiempo real para poder determinar su velocidad  $\dot{y}$ , aceleración  $\ddot{y}$  y sobreaceleración  $y^{(3)}$ .

Al utilizar la derivación numérica como herramienta para obtener la velocidad  $\dot{y}$  a partir de la posición  $y$ , se obtiene la señal derivada mostrada en la Fig. 4.2. El resultado obtenido nos ha servido para descubrir que la medición realizada por el sistema de cámaras Optitrack es sumamente ruidosa. En un intento por calcular

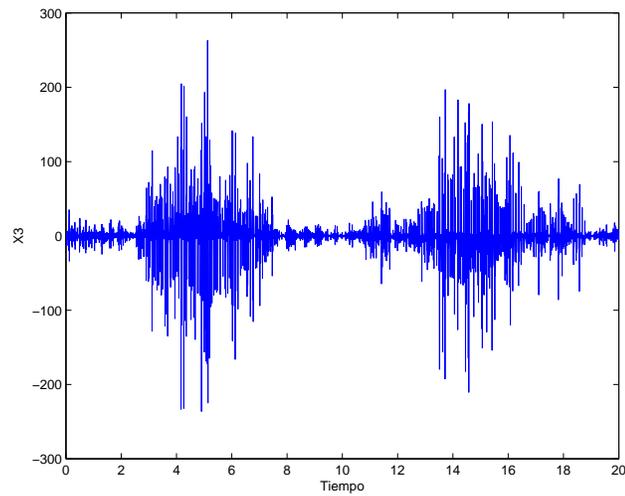


**Figura 4.1.** Desplazamiento del drone sobre el eje Y (estado  $x_1$ )

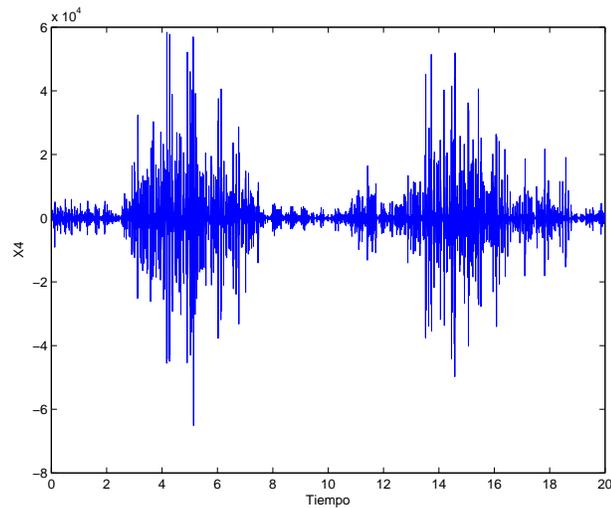
las derivadas de mayor orden de la posición  $y$ , siguiendo el mismo procedimiento, se obtienen los resultados mostrados en las Figs. 4.3 y 4.4, a partir de los cuales es posible concluir que la derivación numérica no es apropiada para estimar los estados  $x_2$ ,  $x_3$  y  $x_4$ .



**Figura 4.2.** Velocidad  $\dot{y}$  (estado  $x_2$ ), por derivación numérica



**Figura 4.3.** Aceleración  $\ddot{y}$  (estado  $x_3$ ), por derivación numérica



**Figura 4.4.** Sobreaceleración  $y^{(3)}$  (estado  $x_4$ ), por derivación numérica

## 4.1. Diferenciador de Levant

En [8] se presenta una interesante aplicación del control por modos deslizantes de orden superior. La idea es utilizar el método como observador para diferenciar una señal. Como ventajas más notables cabe destacar que se consigue un diferenciador que

es exacto en ausencia de ruidos y robusto en presencia de ruidos. Dicho diferenciador, conocido como diferenciador de Levant, es capaz de encontrar estimaciones robustas en tiempo finito de las derivadas con respecto al tiempo de una señal de entrada, cuando su derivada es Lipschitz.

En el citado artículo se presenta el diferenciador de Levant de la siguiente manera.

Sea la señal de entrada  $f(t)$  al diferenciador una función definida en  $[0, \infty)$ . La función  $f(t)$  es una función medible y  $f(t) = f_0(t) + v(t)$ . El primer término es la señal base desconocida  $f_0(t)$  que tiene una derivada con constante de Lipschitz conocida  $L > 0$ . El segundo término corresponde al ruido  $v(t)$  que se adiciona a la señal  $f_0(t)$ , el cual es desconocido, pero se asume acotado. Entonces un diferenciador de  $n$ -ésimo orden está representado por el sistema de ecuaciones

$$\begin{aligned}
v_0 &= -\lambda_0 L^{1/(n+1)} |z_0 - f(t)|^{n/(n+1)} \text{sign}(z_0 - f(t)) + z_1 \\
v_1 &= -\lambda_1 L^{1/n} |z_1 - v_0|^{(n-1)/n} \text{sign}(z_1 - v_0) + z_2 \\
&\vdots \\
v_{n-1} &= -\lambda_{n-1} L^{1/2} |z_{n-1} - v_{n-2}|^{1/2} \text{sign}(z_{n-1} - v_{n-2}) + z_n \\
\dot{z}_0 &= v_0 \\
\dot{z}_1 &= v_1 \\
&\vdots \\
\dot{z}_{n-1} &= v_{n-1} \\
\dot{z}_n &= -\lambda_n L \text{sign}(z_n - v_{n-1})
\end{aligned} \tag{4.2}$$

donde  $z_0$  es la estimación de la función  $f(t)$  y  $z_i$  es la  $i$ -ésima derivada de  $f(t)$ . Los parámetros  $\lambda_i > 0$  son elegidos recursivamente mediante simulación y/o experimentación.

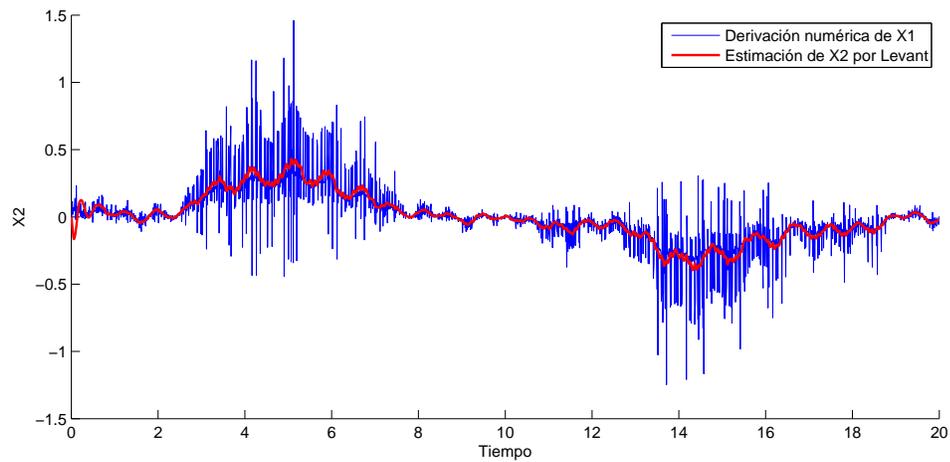
Regresemos al ejemplo anterior donde la Fig. 4.1 muestra el desplazamiento  $y$  ( $x_1$ ) del dron medido desde el sistema de cámaras Optitrack en un experimento en tiempo real. Utilizando un diferenciador de tercer orden se obtiene una estimación de las tres primeras derivadas con respecto al tiempo de dicho desplazamiento  $y$ .

La Fig. 4.5 muestra una comparación entre el estado  $x_2$  obtenido por derivación numérica de  $x_1$  y la estimación de  $x_2$  obtenida mediante el diferenciador de Levant.

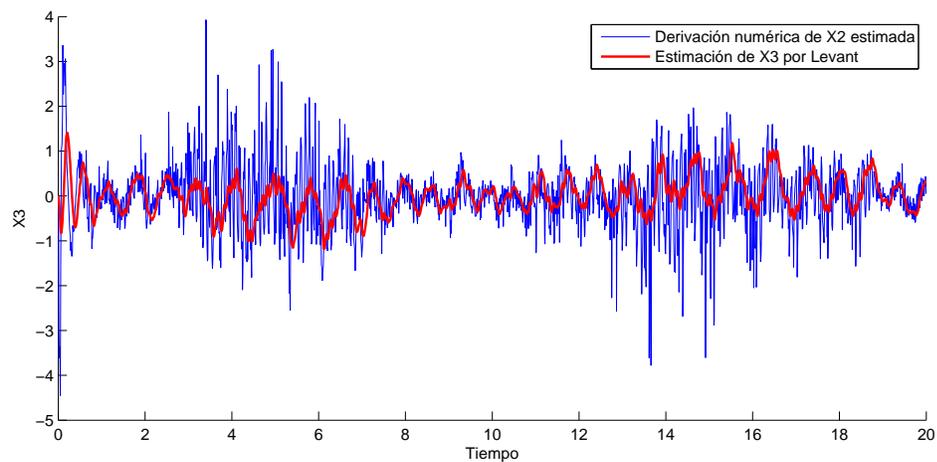
La figura 4.6 hace una comparación entre el estado  $x_3$  obtenido por derivación numérica de la estimación (mediante Levant) de  $x_2$  y la estimación de  $x_3$  obtenida mediante el diferenciador de Levant.

Finalmente, la figura 4.7 hace una comparación entre el estado  $x_4$  obtenido por derivación numérica de la estimación (mediante Levant) de  $x_3$  y la estimación de  $x_4$  obtenida mediante el diferenciador de Levant.

Para nuestro sistema se requiere diseñar tres diferenciadores de Levant: uno de tercer orden para estimar las tres primeras derivadas con respecto al tiempo de  $x_1$ ,



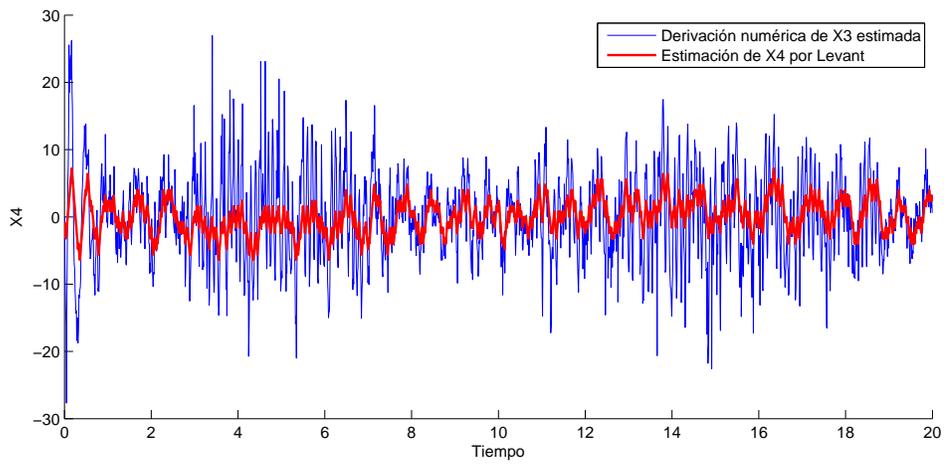
**Figura 4.5.** Estado  $x_2$  obtenido por derivación numérica y por Levant



**Figura 4.6.** Estado  $x_3$  obtenido por derivación numérica y por Levant

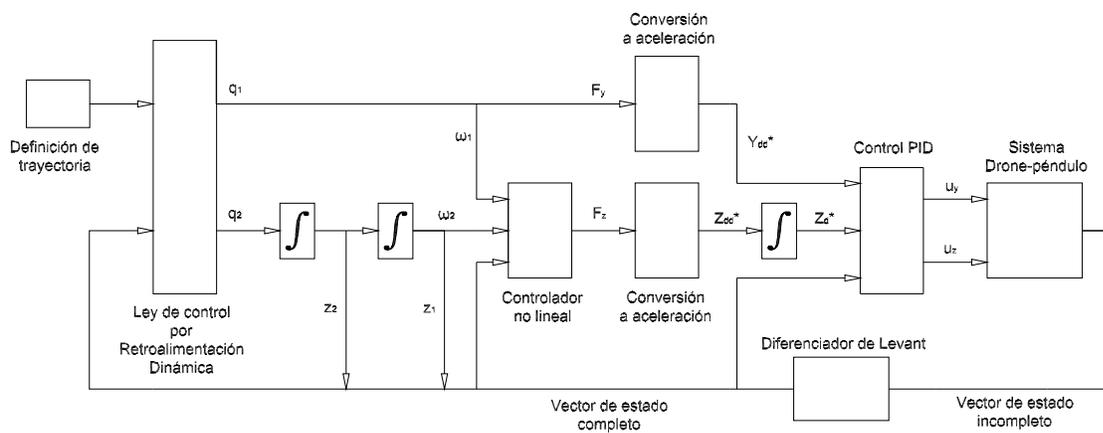
y dos diferenciadores de primer orden para estimar la primera derivada de  $x_5$  y de  $x_7$ .

La Fig. 4.8 muestra el esquema de control desarrollado en el capítulo 3. Nótese que a diferencia de la Fig. 3.8, este nuevo esquema incluye al diferenciador de Levant, mismo que es utilizado para estimar las variables de estado no conocidas. De este modo es posible construir un vector de estado completo con los estados medibles y



**Figura 4.7.** Estado  $x_4$  obtenido por derivación numérica y por Levant

los estimados.



**Figura 4.8.** Esquema de control por Retroalimentación Dinámica - PD, con diferenciador de Levant

# Capítulo 5

## Resultados en simulación y tiempo real

La estrategia de control propuesta en el capítulo 3 es implementada y analizada en este capítulo. El análisis se lleva a cabo mediante experimentos en simulación y en tiempo real.

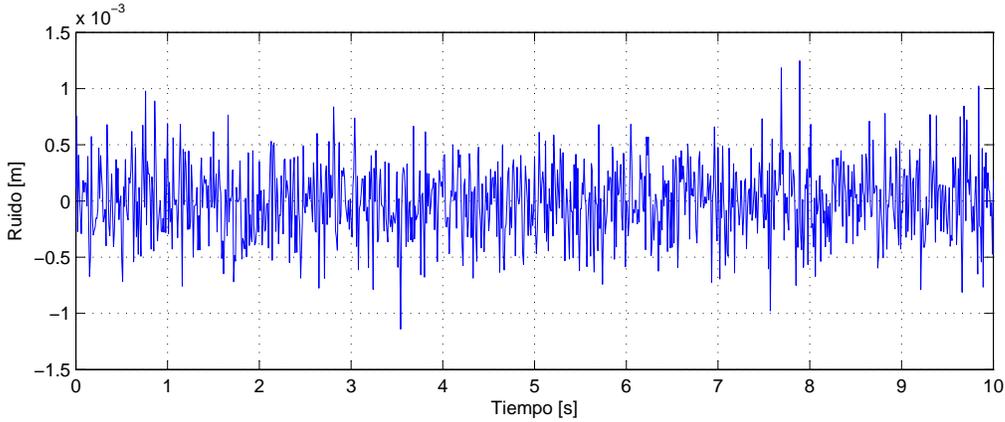
En ambos casos se utilizó con una cuerda con longitud de  $0.773\text{ m}$ , masa  $m_1$  del cuadrirotor de  $0.46\text{ kg}$ , masa  $m_2$  del objeto pendiente de  $0.014\text{ kg}$  y constante de aceleración gravitacional de  $9.81\text{ m/s}^2$ .

Para analizar los resultados obtenidos se graficarán los errores de seguimiento de trayectoria (dados por  $e = h_d - h$ ) y las señales de control  $u_y$  y  $u_z$ .

### 5.1. Experimentos en simulación

A continuación se consideran dos casos en simulación. En uno de ellos la asume la posibilidad de medir directamente del sistema sus ocho estados y aplicar las leyes de control diseñadas en el capítulo 3.

En el segundo caso, se considera que solo los estados  $x_1$ ,  $x_5$  y  $x_7$  pueden medirse y se hará uso del diferenciador de Levant propuesto en el capítulo 4 para estimar el resto de los estados. Se sumará a dichas señales medibles  $x_1(t)$ ,  $x_5(t)$  y  $x_7(t)$  una señal ruidosa como la mostrada en la Fig. 5.1 con el objetivo de aproximar la simulación al problema real, en el cual la medición de los estados presenta ruido similar al mencionado. Con lo anterior se pretende comprobar la robustez del diferenciador de Levant. Una vez construido un vector de estado completo se aplicarán las mismas leyes de control bajo el principio de equivalencia cierta.



**Figura 5.1.** Ruido blanco añadido a la medición de los estados

### 5.1.1. Regulación del objeto pendiente en un punto en el plano

Como primer experimento en simulación considérese la regulación de la posición del objeto pendiente en un punto en el plano  $YZ$ . Supóngase que el sistema tiene las siguientes condiciones iniciales:

$$\begin{aligned} x_1 = -2, \quad x_2 = -0.09, \quad x_3 = 0.4, \quad x_4 = 0.5 \\ x_5 = 1.2, \quad x_6 = 2, \quad x_7 = -0.2, \quad x_8 = 0.3 \end{aligned}$$

Sea el origen del plano  $YZ$  el punto en donde deseamos regular la carga pendiente. De la simulación se obtienen las gráficas mostradas en las Fig. 5.2 y 5.3.

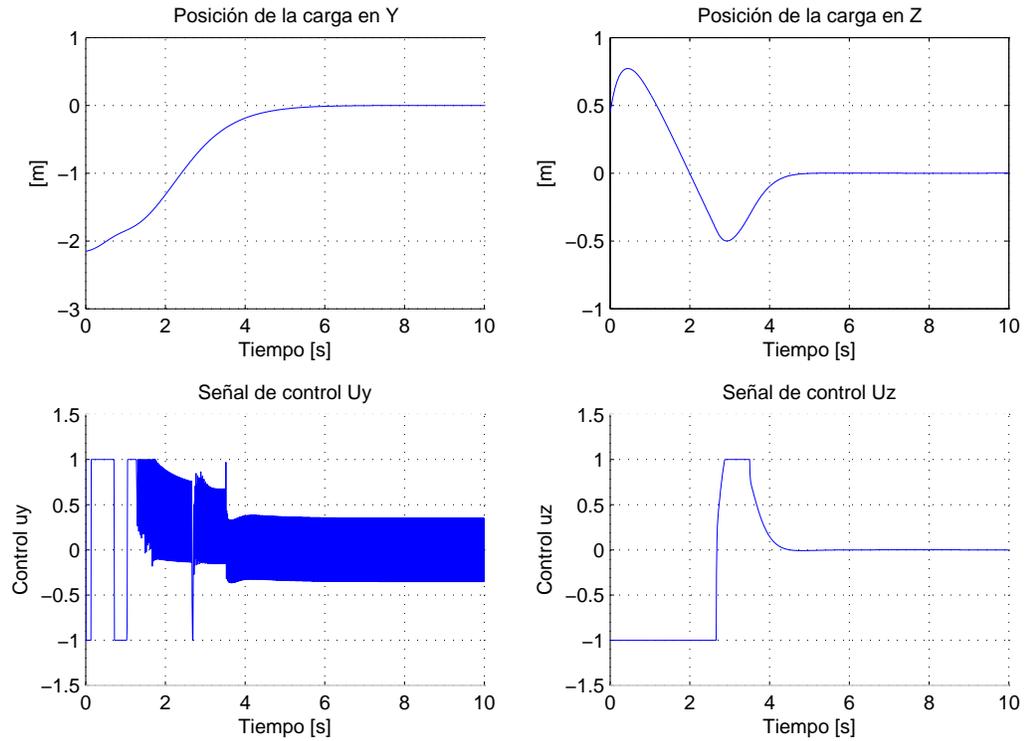
### 5.1.2. Seguimiento de trayectoria

Se desea ahora seguir una trayectoria que sea una función suave del tiempo. Dicha trayectoria consiste de 6 etapas como se muestra en la Fig. 5.4.

La trayectoria ABCDEFG inicia en el punto A en  $t_A = 0$ . Los instantes de tiempo correspondientes a los puntos B, C, D, E, F y G son  $t_B$ ,  $t_C$ ,  $t_D$ ,  $t_E$ ,  $t_F$  y  $t_G$ , respectivamente. Nótese que los puntos A y G tienen las mismas coordenadas cartesianas y marcan el inicio y el final de la trayectoria.

La trayectoria deseada se construye mediante polinomios que son funciones suaves del tiempo:

$$\begin{aligned} h_{1d} &= h_{1_0} + \Delta_D \psi \\ h_{2d} &= h_{2_0} + \Delta_L \psi \end{aligned}$$



**Figura 5.2.** Experimento en simulación: Regulación de la carga en el origen con medición directa de los ocho estados

donde

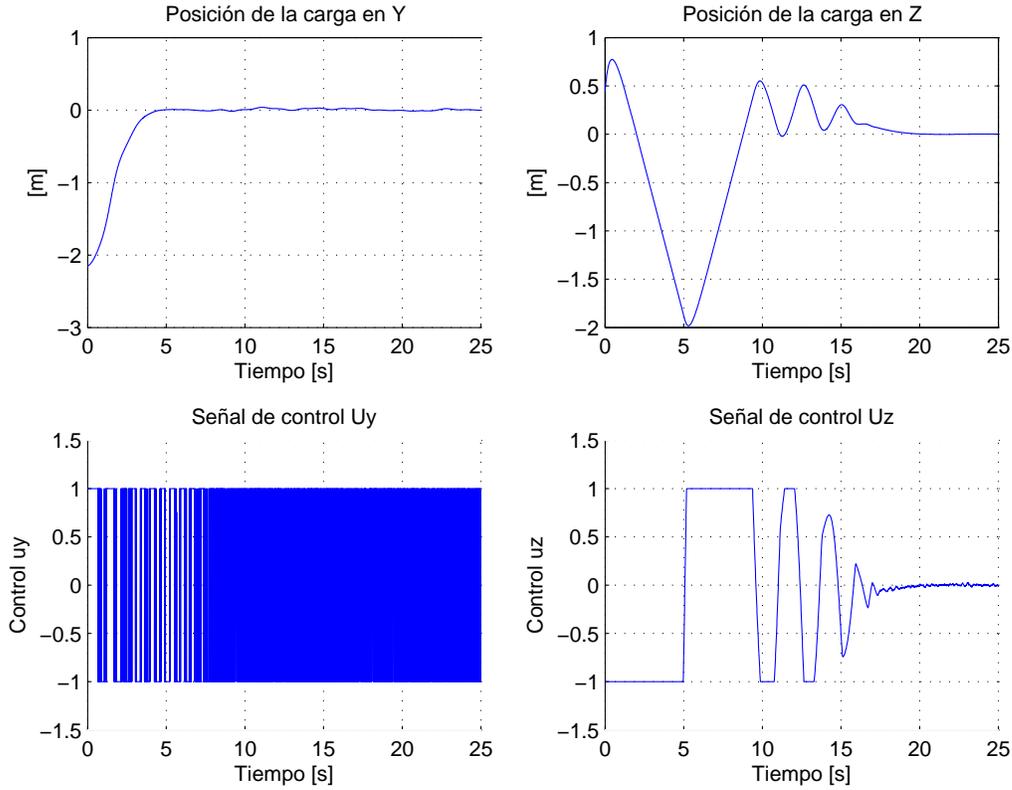
$$\psi = \frac{\Delta_t^5}{\Delta_T^5} \left( 126 - \frac{420\Delta_t}{\Delta_T} + \frac{540\Delta_t^2}{\Delta_T^2} - \frac{315\Delta_t^3}{\Delta_T^3} + \frac{70\Delta_t^4}{\Delta_T^4} \right)$$

$$\Delta_t = t - t_0$$

$$\Delta_T = t_f - t_0$$

En las expresiones anteriores,  $h_{1_0}$  y  $h_{2_0}$  son las condiciones iniciales para  $h_{1_d}$  y  $h_{2_d}$ , respectivamente. Asimismo,  $\Delta_D$  y  $\Delta_L$  son las distancias totales recorridas en las direcciones horizontal y vertical. Finalmente,  $t_0$  y  $t_f$  son los tiempos inicial y final de la trayectoria que se recorre desde la posición de inicio hasta el punto señalado como objetivo.

Para definir las longitudes de la trayectoria se considera:



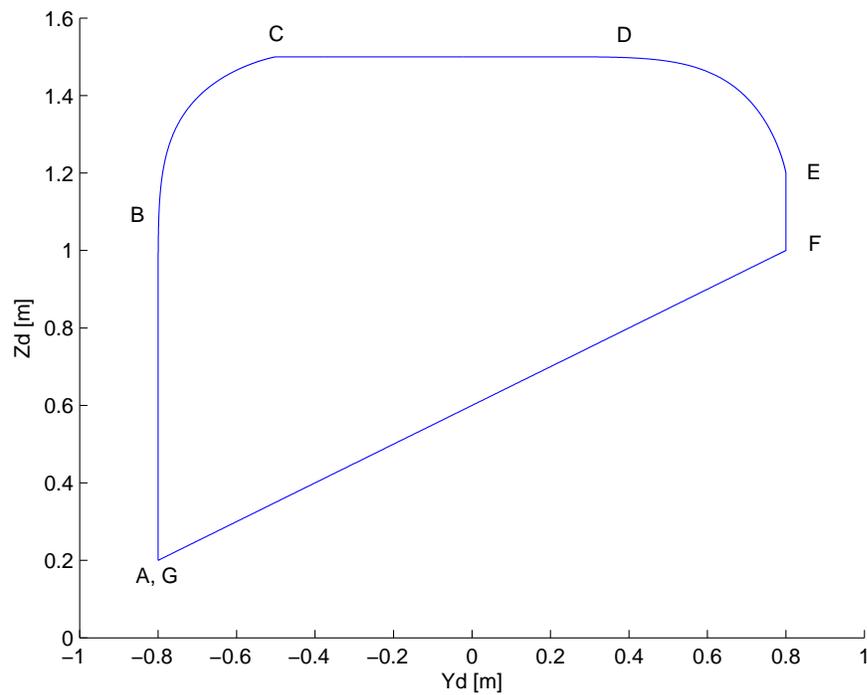
**Figura 5.3.** Experimento en simulación: Regulación de la carga en el origen con estimación de algunos estados mediante diferenciador de Levant

$$\begin{aligned} \Delta_{D_{AB}} &= 0, & \Delta_{D_{AC}} &= 0.3, & \Delta_{D_{CE}} &= 1.3, & \Delta_{D_{EF}} &= 0, & \Delta_{D_{AF}} &= 1.6 \\ \Delta_{L_{AC}} &= 1.3, & \Delta_{L_{CD}} &= 0, & \Delta_{D_{CE}} &= 0.3, & \Delta_{L_{EF}} &= 0.2, & \Delta_{L_{AF}} &= 0.8 \\ & & & & h_{1_0} &= -0.8, & h_{2_0} &= 0.2 \end{aligned}$$

en metros, mientras que los instantes de tiempo en segundos de cada punto son:

$$t_A = 0, \quad t_B = 10, \quad t_C = 20, \quad t_D = 30, \quad t_E = 40, \quad t_F = 45, \quad t_G = 55$$

Una vez definida la trayectoria deseada, se establecen las condiciones iniciales de los



**Figura 5.4.** Trayectoria deseada para la carga

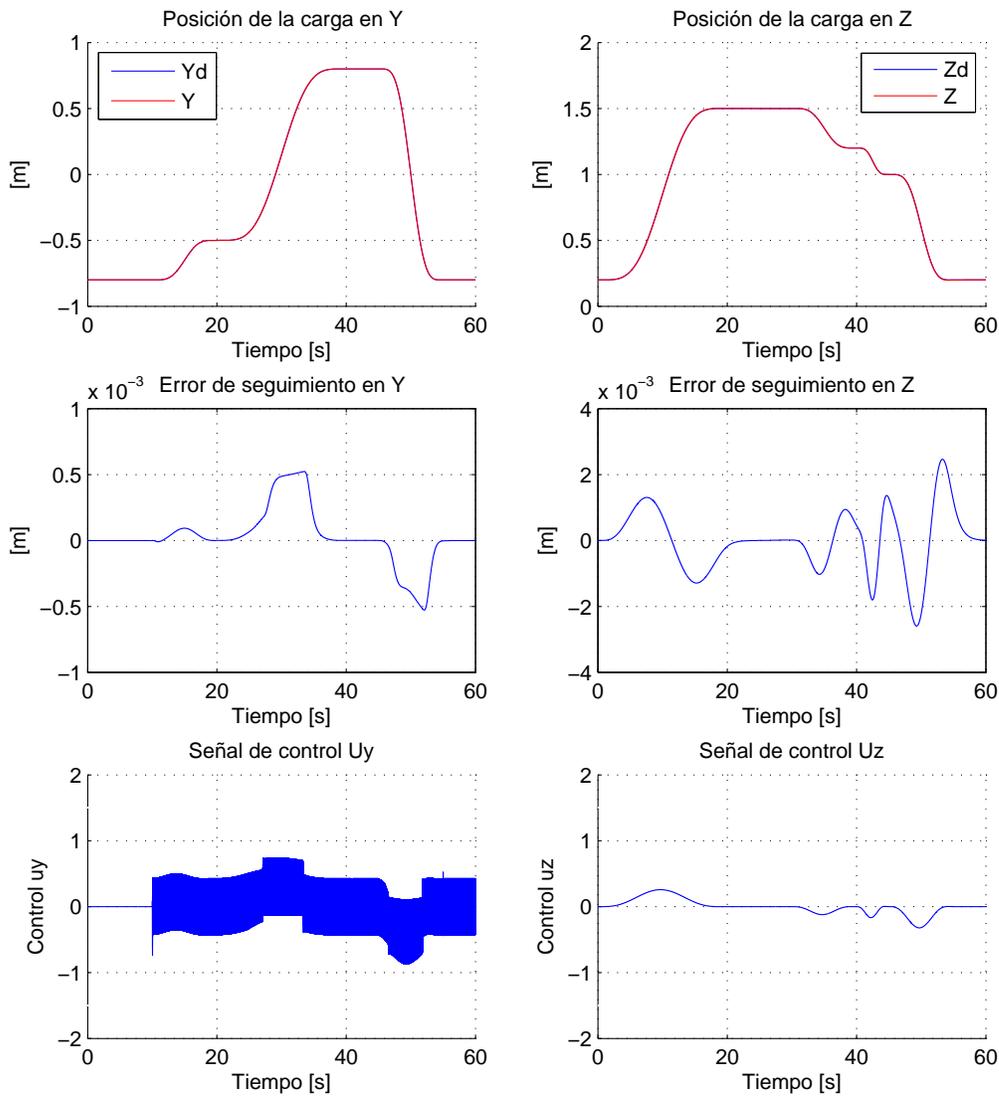
estados del sistema:

$$\begin{aligned} x_1 &= -0.8, & x_2 &= 0, & x_3 &= 0, & x_4 &= 0 \\ x_5 &= 0.973, & x_6 &= 0, & x_7 &= 0, & x_8 &= 0 \end{aligned}$$

De la simulación se obtienen las gráficas mostradas en las Fig. 5.5 y 5.6.

## 5.2. Experimentos en tiempo real

En los experimentos en tiempo real solo se presentan los resultados obtenidos cuando los estados  $x_2$ ,  $x_3$ ,  $x_4$ ,  $x_6$  y  $x_8$  fueron estimados con ayuda del diferenciador de Levant.



**Figura 5.5.** Experimento en simulación: Seguimiento de trayectoria con medición directa de los ocho estados

### 5.2.1. Regulación del objeto pendiente en un punto en el plano

En el primer experimento en tiempo real se realizó la regulación de posición del objeto pendiente sobre un punto en el plano  $YZ$  con coordenadas cartesianas  $(0,0,$

1.0). Las condiciones iniciales del sistema fueron:

$$\begin{aligned}x_1 &= 0.44, & x_2 &= 0.04, & x_3 &= -0.09, & x_4 &= -0.99 \\x_5 &= 1.5, & x_6 &= 0.1, & x_7 &= 0.04, & x_8 &= 0.15\end{aligned}$$

Adicionalmente, en cada uno de los instantes  $t = 10.09s$ ,  $t = 31.41s$  y  $t = 54.32s$  se aplicó un impulso (cuya dirección y magnitud se desconocen) al objeto pendiente. Dichos impulsos actúan como perturbaciones para alejar al objeto del punto de referencia. Lo anterior para comprobar el desempeño del control diseñado. Las gráficas de la Fig. 5.7 muestran los resultados obtenidos en tiempo real para la regulación en un punto en el plano.

En un experimento adicional en tiempo real se analizó el comportamiento del sistema cuando éste no fue controlado por ninguna ley de control; es decir, cuando  $u_y$  y  $u_z$  fueron ambos cero. Inicialmente el sistema se encontró en equilibrio (vuelo estacionario) con la carga posicionada en el punto  $(0.05, 0.50)$ . Siguiendo el mismo procedimiento que en el experimento anterior, en  $t = 4.32s$  se aplicó un impulso (con dirección y magnitud desconocidas) al objeto a fin de perturbarlo. El resultado obtenido se muestra en la Fig. 5.8. Obsérvese que cuando no se ejerce la ley de control en el sistema el tiempo de oscilación del péndulo es significativamente mayor que cuando el sistema es controlado.

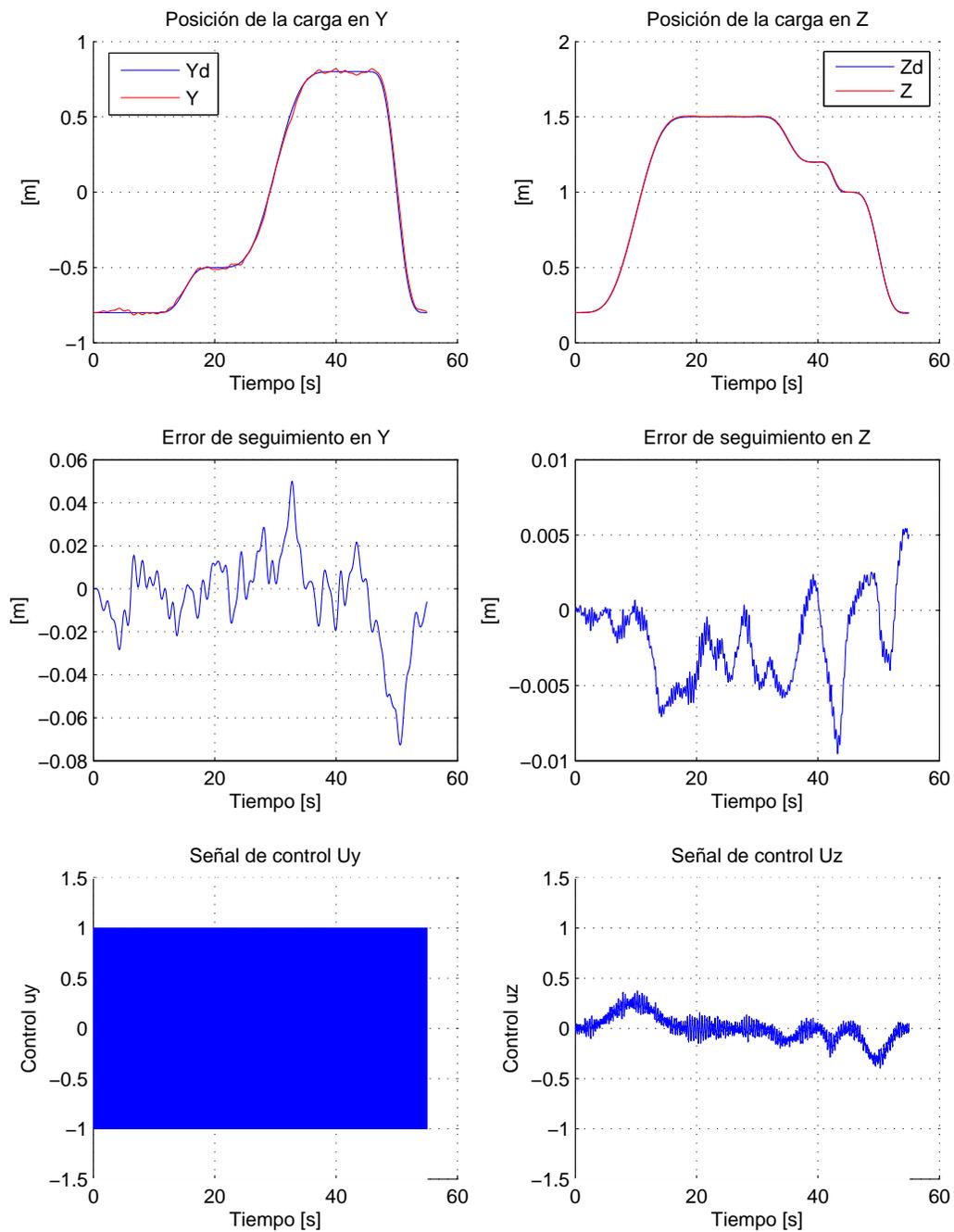
### 5.2.2. Seguimiento de trayectoria

La trayectoria propuesta para el experimento en tiempo real es la misma que en el experimento en simulación.

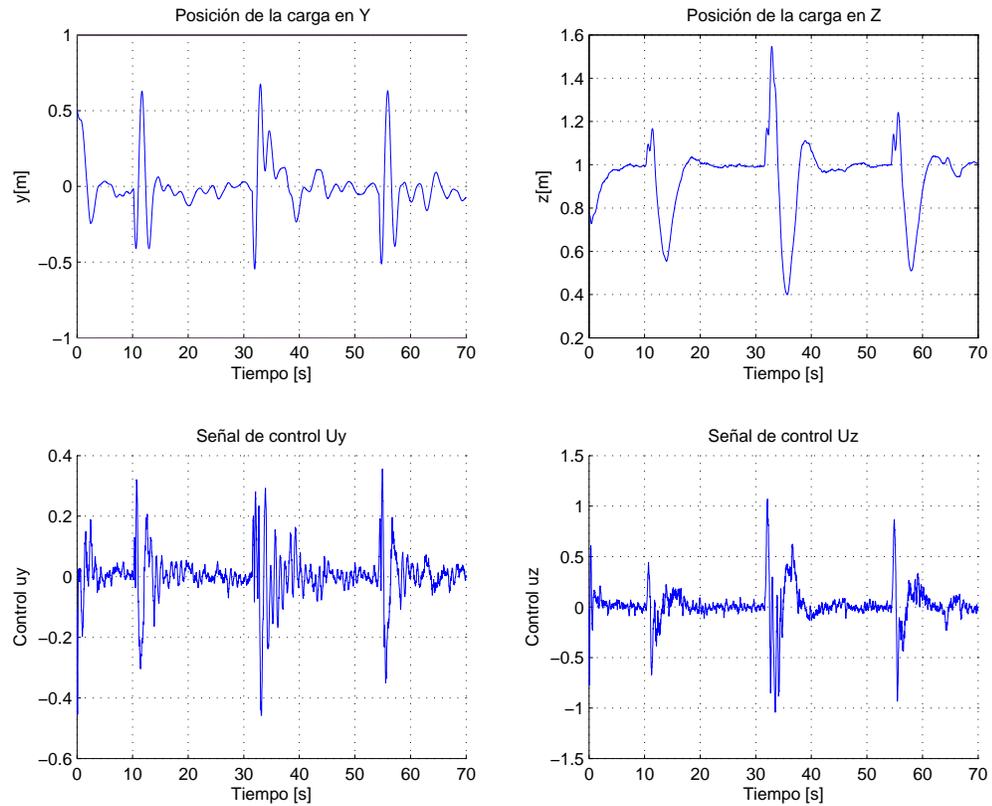
Las condiciones iniciales fueron:

$$\begin{aligned}x_1 &= -0.85, & x_2 &= -0.13, & x_3 &= 0.21, & x_4 &= 1.23 \\x_5 &= 1, & x_6 &= -0.14, & x_7 &= 0.23, & x_8 &= -0.97\end{aligned}$$

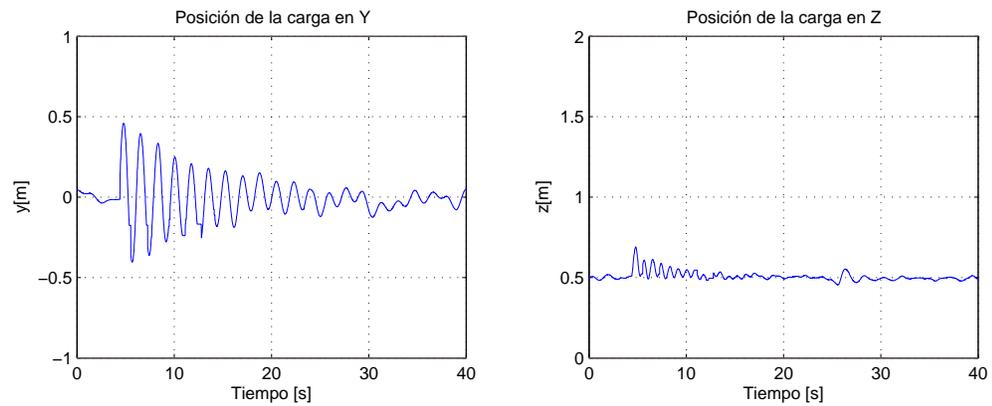
Las Fig. 5.9 y 5.10 muestran los resultados obtenidos.



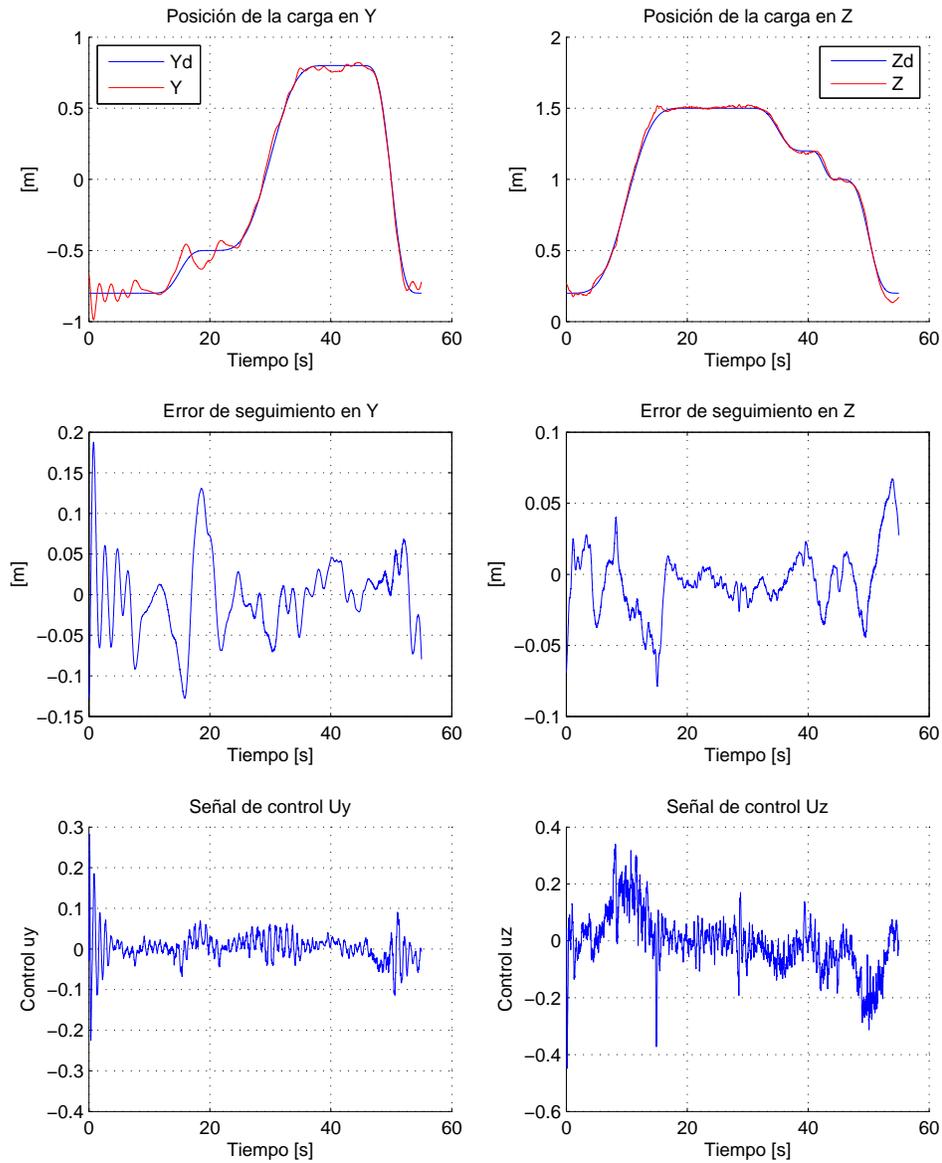
**Figura 5.6.** Experimento en simulación: Seguimiento de trayectoria con estimación de algunos estados mediante diferenciador de Levant



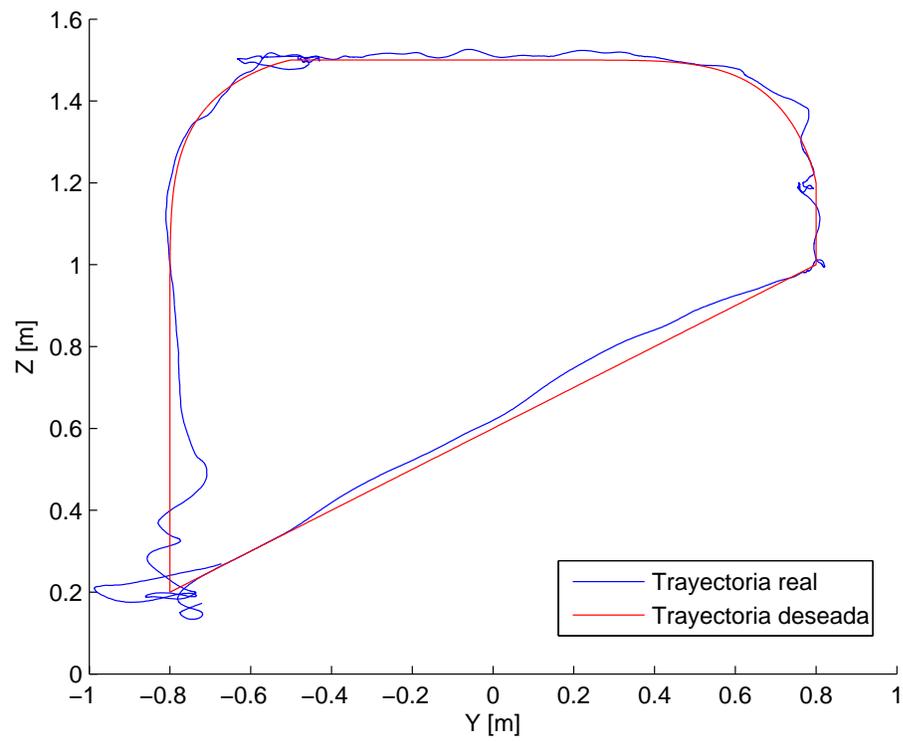
**Figura 5.7.** Experimento en tiempo real: Regulación de la carga sobre un punto en el plano, experimento en tiempo real



**Figura 5.8.** Experimento en tiempo real: Impulso aplicado al péndulo sin control



**Figura 5.9.** Experimento en tiempo real: Seguimiento de trayectoria



**Figura 5.10.** Experimento en tiempo real: Trayectoria deseada vs real



# Capítulo 6

## Conclusiones

Hoy en día, la industria requiere utilizar grúas viajeras más eficientes para el desplazamiento de material de un lugar a otro. En un ambiente industrial se requiere de procesos rápidos, exactos y eficientes con el propósito de incrementar la tasa de producción, el control de calidad, así como sus utilidades; por lo tanto, se crea la necesidad de implementar grúas viajeras que trasladen materiales más rápido, con mayor precisión y seguridad. En consecuencia, el diseño, construcción y control de grúas viajeras son problemas que requieren cada vez de más y mejores soluciones.

Con el fin de aportar una solución al problema de traslado de material, desde un nuevo enfoque, se implementó un controlador para una grúa viajera aérea de tres grados de libertad con dos entradas de control (sistema subactuado), que sirvió como plataforma experimental para el desarrollo de nuevas leyes de control. En particular, en este trabajo se diseñó una ley de control para el seguimiento asintótico o exacto de trayectorias para el desplazamiento de la carga de la grúa.

Para la construcción del prototipo experimental se parte de un cuadrirotor comercial del cual se sujeta una carga a través de una cuerda, misma que puede balancearse libremente en el aire.

Luego, se desarrolló matemáticamente un modelo dinámico para la grúa viajera con el propósito de estudiar su comportamiento, propiedades y características. Sin embargo, el cuadrirotor utilizado en este proyecto de tesis cuenta con un controlador interno de su fabricante y da al usuario la posibilidad de dirigirlo mediante señales de control transmitidas inalámbricamente. Dicho controlador es de naturaleza hermética, es decir, el usuario no tiene acceso a todos los estados del sistema. Por lo anterior se propuso modelar dicha aeronave como un conjunto de sistemas lineales tipo “caja negra” en la que se desconoce la dinámica entre las velocidades de rotación de las hélices y las posiciones, velocidades o aceleraciones del cuerpo. Así, el modelo dinámico obtenido es una aproximación al modelo real y, por lo tanto, existen incertidumbres matemáticas en dicho modelo.

Lo anterior define entonces el problema principal de esta tesis: el control de un sistema subactuado y con incertidumbres matemáticas en su modelo dinámico.

Sobre la base de este modelo se estudió y analizó el comportamiento de la grúa

mediante simulación numérica. A pesar de la inexactitud del modelo dinámico obtenido, el comportamiento del sistema en simulación es bastante aproximado al comportamiento del sistema real.

Para resolver el problema de control del modelo dinámico inexacto del sistema, se propuso descomponer dicho modelo en dos partes: la primera de ellas consistió en un submodelo matemático sin incertidumbres matemáticas, es decir, la parte exacta del modelo general; mientras que la segunda parte consistió en un submodelo matemático que incluye todas las dinámicas no conocidas del sistema (incertidumbres matemáticas). Entonces, se diseñó un controlador para cada submodelo matemático, que en conjunto forman un controlador jerárquico de dos etapas.

En la primera etapa consideró que el cuadrirotor y objeto pendiente son masas inertes sin control propio y que la masa de la aeronave podría ser desplazada estratégicamente en el espacio mediante fuerzas de empuje, de tal modo que la masa pendiente sea conducida en la dirección deseada. Este planeamiento define un submodelo dinámico sin incertidumbres matemáticas. Las fuerzas de empuje que actúan sobre la masa del cuadrirotor fueron calculadas en esta etapa mediante una linealización por retroalimentación dinámica.

En la segunda etapa se tomó en cuenta el controlador interno que posee el cuadrirotor, mismo que permite al usuario manipular a la aeronave mediante señales de control inalámbricas. Así, el usuario envía a la aeronave el ángulo de referencia deseado para el alabeo, así como la velocidad de referencia deseada de ascenso/descenso; luego, el controlador del cuadrirotor se encarga de producir dichos valores de referencia deseados. Este sistema define el segundo submodelo matemático del cual sólo se tiene una aproximación. Entonces, se diseñó un control tipo proporcional integral derivativo (PID) que determina las señales de control adecuadas hacia el cuadrirotor y logra que su ángulo de alabeo y velocidad de ascenso/descenso produzcan desplazamientos equivalentes al que producirían las fuerzas de empuje obtenidas en la primera etapa. Cabe mencionar que el control de los ángulos de cabeceo y guiñada no fue implementado, manteniendo dichas posiciones angulares en un valor constante.

Ambas leyes de control, creadas en las dos etapas mencionadas, funcionan conjuntamente de manera jerárquica a fin de lograr el seguimiento de una trayectoria deseada para el objeto pendiente, o bien, su regulación en un punto en el espacio.

Es importante mencionar que el esquema de control anterior, basado en la descomposición del modelo dinámico general en dos submodelos matemáticos controlados independientemente cada uno por un controlador distinto, es una aportación original de este trabajo.

Una ventaja del esquema de control jerarquizado es que al partir de submodelos dinámicos más sencillos se requiere de un tiempo (o potencia) de cálculo menor, ya que la aplicación de un algoritmo de extensión dinámica para linealizar el modelo dinámico general del sistema representaría expresiones matemáticas complejas y sumamente extensas, lo que demandaría mayor tiempo (o potencia) para su procesamiento.

Los esquemas de control desarrollados requieren del conocimiento completo del

estado. La medición directa de algunos estados del sistema en tiempo real representó una fuerte dificultad técnica. Sin embargo, a partir de la definición de las ocho variables de estado del sistema grúa-péndulo, se comprueba que cinco de ellas son derivadas sucesivas con respecto al tiempo de las tres variables restantes. Por lo anterior, se optó por el empleo del Diferenciador de Levant para obtener numéricamente una aproximación de las variables de estado no medibles del sistema. Dicho diferenciador es una aplicación del control por modos deslizantes de orden superior. La idea fue utilizar el método como observador para diferenciar una señal. Como ventajas más notables cabe destacar que se consiguió un diferenciador que fue robusto ante la presencia de ruidos en la señal de entrada (estados medibles).

Finalmente, se evaluó el desempeño del esquema de control diseñado. Dicha evaluación se llevó a cabo mediante experimentos en simulación numérica y tiempo real tanto para el seguimiento de una trayectoria en el plano vertical para la carga pendiente, como para su regulación en un punto en el plano.

Para el caso de la simulación, se estudiaron dos casos. Uno de ellos consideró la posibilidad de medir directamente del sistema sus ocho variables de estado. En el otro caso solo se midió directamente tres de ellas añadiéndoles una señal ruidosa, tal como sucede en la medición en tiempo real; luego, las cinco restantes fueron obtenidas mediante el uso del Diferenciador de Levant.

Por otra parte, en los experimentos en tiempo real sólo se estudió el caso en el que se miden directamente tres de las ocho variables de estado del sistema y obteniendo una aproximación de las cinco restantes mediante el diferenciador de Levant.

En ambos experimentos, simulación y tiempo real, el esquema de control diseñado funciona correctamente y muestra un buen desempeño.

Los resultados del desempeño del controlador en dichos experimentos se mostraron mediante gráficas donde se observó que el error de seguimiento convergió rápidamente a cero.

Dados los resultados anteriores y considerando que el esquema de control que se diseñó en este trabajo nunca se había implementado en tiempo real, es posible concluir que este controlador es una importante aportación para el problema de control de grúas viajeras, además de que provee robustez ante perturbaciones presentes durante la operación; tales como la desviación del objeto pendiente debido a corrientes de aire, entre otras.

Algunas perspectivas y trabajos futuros relacionados con el presentado en esta tesis son las siguientes:

Extender el controlador aquí diseñado para permitir la realización de operaciones en el espacio  $XYZ$ , ya que aquí se asume que el sistema se desenvuelve en el plano vertical  $YZ$ , siendo éste un escenario de trabajo con demasiadas limitaciones en cuanto a espacio físico de trabajo.

Proponer nuevas trayectorias deseadas para la carga pendiente, a fin de evaluar a plenitud el desempeño de la grúa a lo largo de todo su espacio de operación.

Implementación de distintos observadores para estimar las variables de estado no medibles del sistema, mismos que construyan aproximaciones de dichos estados

convergiendo a los estados reales en menor tiempo.

Obtener una mejor aproximación del modelo dinámico del cuadrirotor, eliminando el mayor número de incertidumbres matemáticas posible.

Considerar la complejidad del modelo matemático de una cuerda estándar que se basa en la formulación de ecuaciones diferenciales parciales (PDEs).

# Bibliografía

- [1] L.R. García Carrillo, A.E. Dzúl López, R. Lozano, and C. Pégard, *Quad rotorcraft control: Vision-based hovering and navigation*, Springer-Verlag, 2013.
- [2] P.J. Bristeau, F. Callou, D. Vissière, and N. Petit, “The navigation and control technology inside the Ar. Drone micro UAV”, *18th IFAC World Congress*, Milano, Italy, pp. 1477-1484, 2011.
- [3] T. Krajník, V. Vonásek, Daniel Fišer, and J. Faigl, “AR-Drone as a Platform for Robotic Research and Education”, *Research and Education in Robotics EU-ROBOT*, International Conference, Prague, Czech Republic, pp. 172-186, 2011.
- [4] W. Aguilar, C. Angulo, R. Costa-Castelló, y L. Molina, “Control autónomo de cuadricópteros para seguimiento de trayectorias”, *Congreso de Ciencia y Tecnología, Revista Digital Congreso de Ciencia y Tecnología: Memorias, Sesiones Técnicas, Sangolquí.*, pp. 140-144, 2014.
- [5] J. Mauri Alarcon, *Disseny d'un controlador per l'AR.Drone 2.0*, Master thesis, Departament d'Enginyeria de Sistemes, Automàtica i Informàtica Industrial, Universitat Politècnica de Catalunya, Barcelona, Espagne, 2014.
- [6] M. Méndez Martínez, *Control de vuelo en configuración Líder-Seguidor para aeronaves tipo cuadrirotor*, tesis de Maestría, Dept. Ing. Eléctrica, Sección de Mecatrónica, Centro de Investigación y de Estudios Avanzados del Instituto Politécnico Nacional, México D.F., 2014.
- [7] A. Isidori, *Nonlinear Control Systems*, 3rd ed., Springer-Verlag, 1995.
- [8] A. Levant, “Higher-order sliding modes, differentiation and output-feedback control”, *International Journal of Control*, Vol. 76: 9-10, pp. 924-941, 2003.
- [9] Ros.org: Powering the world's robots. [www.ros.org](http://www.ros.org) [consultado: Noviembre 2014].
- [10] ROS installation made easy. <http://nootrix.com/2012/05/ros-installation/> [consultado: Noviembre 2014].

- [11] Up and flying with the AR-Drone and ROS. <http://robohub.org/up-and-flying-with-the-ar-drone-and-ros-getting-started/> [Consultado: Noviembre 2014]
- [12] ROS/Tutorials. <http://wiki.ros.org/ROS/Tutorials> [Consultado: Diciembre 2014]

# Apéndice A

## Artículo publicado

A. J. Hernández Mejía y E. Aranda-Bricaire, “Control de posición de un objeto pendiente de un cuadrirotor”, *Memorias del XVII Congreso Mexicano de Robótica COMRob 2015*, Los Cabos, México, del 11 al 13 de Noviembre de 2015.