



**CENTRO DE INVESTIGACIÓN Y DE ESTUDIOS AVANZADOS
DEL INSTITUTO POLITÉCNICO NACIONAL**

UNIDAD ZACATENCO
DEPARTAMENTO DE INGENIERÍA ELÉCTRICA
SECCIÓN DE MECATRÓNICA

**Control de vuelo en Configuración Líder-seguidor para
Aeronaves tipo Cuadrirotor**

Tesis que presenta:
Mauricio Méndez Martínez

Para obtener el grado de:
Maestro en Ciencias

En la especialidad de:
Ingeniería Eléctrica

Director de la Tesis:
Dr. Eduardo Aranda Bricaire

Agradecimientos

A: *Por*

A mi madre Rosa María: *Por su apoyo, comprensión y confianza. Le agradezco por los esfuerzos que realizo para que pudiera tener una buena educación a pesar de las limitaciones y etapas difíciles que pasamos. Espero dar siempre lo mejor de mí y no defraudarla, trabajar duro y con dedicación para que se sienta orgullosa.*

Al Dr. Eduardo Aranda: *Por sus consejos, su apoyo y sus críticas para lograr 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, sus consejos y por compartir sus conocimientos y experiencias conmigo. Por confiar en mí y darme la oportunidad de realizar mis estudios de maestría en la sección de mecatrónica.*

A mis amigos y compañeros del CINVESTAV: *Por compartir su amistad conmigo y los buenos momentos que pasamos juntos.*

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

Resumen

El presente trabajo describe el diseño de algunas estrategias de control empleadas para alcanzar una formación geométrica y seguimiento de trayectorias en un sistema multi-agente conformado por dos cuadrirotos. Dichas estrategias de control se proponen con base en funciones de transferencia estimadas a partir de la respuesta del sistema a una señal pulso unitario correspondiente a cada variable de control. De esta forma, se busca modelar a cada cuadrirotor como una “caja negra” en la que se desconoce la relación matemática entre las velocidades de rotación de las hélices y las posiciones, velocidades o aceleraciones del mismo.

Las estrategias de control empleadas son descentralizadas, ya que ningún agente cuenta con conocimiento total del sistema. Se desarrollan dos controladores. El primer controlador se implementa en la aeronave líder para que esta realice el seguimiento de una trayectoria predeterminada a través del espacio. El segundo controlador tiene por objetivo que el agente seguidor mantenga una posición relativa deseada con respecto al agente líder. El segundo controlador se prealimenta con la posición y velocidad del agente líder y se retroalimenta con la posición y velocidad del agente seguidor.

Las estrategias de control diseñadas son implementadas con el uso del lenguaje C++ en el sistema operativo LINUX en su distribución UBUNTU empleando la API AR.Drone SDK 2.0.1. Se validan de forma experimental en una plataforma compuesta de tres partes principales. La primera parte consta de dos cuadrirotos AR.Drone 2.0 producidos por la empresa Parrot, la segunda parte consiste en un sistema de 12 cámaras Flex 13 producidos por la empresa *Natural Point* empleado para determinar las posiciones de los cuadrirotos y la última parte es el sistema de comunicación y control.

Abstract

This paper describes the design of control strategies used to perform a geometric formation and path tracking in a multi-agent system composed of two quadrotors. These control strategies are proposed based on transfer functions estimated from the corresponding system response to a unit pulse signal to each control variable. In this way, we attempt to model each quadrotor as a **black box** in which the mathematical relationship between the angular velocities of the propellers and the positions, velocities and accelerations of the aerial vehicle are unknown.

The applied control strategies are decentralized, since no agent has global knowledge of the environment. Two controllers are developed. The first controller is implemented in the leading aircraft to follow a predetermined path through space. The second controller is intended that the follower agent maintains a desired relative position with respect to the leading agent. The second controller is fed with the position and the velocity of the leading agent and feedback with the follower agent's position and velocity.

The designed control strategies are implemented using the C++ language in the operating system Linux, UBUNTU distribution using the AR.Drone SDK 2.0.1 API. They are validated experimentally in an experimental platform composed of three main parts. The first part is composed of two AR.Drone 2.0 quadrotors produced by Parrot, the second part consists in a system composed of 12 Flex 13 cameras produced by the company Natural Point used to determine the position and orientation of the aircrafts and the last part are the control and communication systems.

Contenido

Lista de figuras	XI
Lista de tablas	XII
1. Introducción	1
1.1. Sistemas multi-agente.	1
1.2. Vehículos aéreos no tripulados.	2
1.3. Estado del arte.	4
1.4. Objetivos de la tesis.	6
1.4.1. Objetivo general.	6
1.4.2. Objetivos específicos.	6
1.5. Aportaciones de la tesis.	7
1.6. Organización de la tesis.	7
2. Planteamiento del problema	9
2.1. Modelo dinámico del Cuadrirotor	9
2.2. Modelo del AR.Drone 2.0	11
2.3. Formación orientada al líder.	18
2.4. Estimación de la velocidad.	19
2.5. Lazos de control.	22
2.5.1. Ley de control para el agente líder.	23
2.5.2. Ley de control para el agente seguidor.	24
2.6. Simulaciones numéricas	24
2.7. Simulación de vuelo individual.	24
2.8. Simulación de vuelo simultaneo en configuración líder-seguidor	27
3. Experimentos en tiempo real	31
3.1. Plataforma experimental.	31
3.1.1. El AR.DRONE 2.0	32
3.1.2. Sistema de posicionamiento	34
3.1.3. Sistema de control y comunicación	35

3.2. Experimentos en tiempo real	36
3.2.1. Vuelo individual	37
3.2.2. Vuelo simultaneo con controladores desacoplados	40
3.2.3. Vuelo simultaneo en configuración líder-seguidor	44
4. Conclusiones y trabajo futuro	49
4.1. Conclusiones.	49
4.2. Trabajo a Futuro.	50
Bibliografía	51
A. Diagramas a bloques	55
A.0.1. Bloque de control	57
A.0.2. Bloque AR.Drone 2.0	57
A.1. Diagrama a bloques para simulación de vuelo individual.	59
A.2. Diagrama a bloques para simulación de vuelo simultaneo en configuración líder-seguidor	59

Lista de figuras

1.1. Modelo de aeronave Harrier	3
1.2. Operación del cuadrirotor	4
2.1. Diagrama de un AR.Drone 2.0 en el espacio	10
2.2. Diagrama entradas / salidas del AR.Drone 2.0	12
2.3. Diagrama entradas / salidas extendido del AR.Drone 2.0	12
2.4. Diagrama de cuerpo libre en estado de flotación	13
2.5. Diagrama de cuerpo libre en estado de flotación con θ modificado	13
2.6. Desplazamientos registrados para una entrada u_z constante	15
2.7. Desplazamientos registrados para una entrada u_ψ constante	15
2.8. Desplazamientos registrados para una entrada u_θ constante	16
2.9. Desplazamientos registrados para una entrada u_ϕ constante	16
2.10. Pulso unitario	17
2.11. Respuestas a entrada pulso	17
2.12. Formación orientada al líder	18
2.13. Señal de prueba	21
2.14. Derivadas estimadas	22
2.15. Diagrama de control	23
2.16. Desplazamiento en x para simulación de vuelo individual	25
2.17. Desplazamiento en y para simulación de vuelo individual	25
2.18. Desplazamiento en z para simulación de vuelo individual	26
2.19. Desplazamiento en ψ para simulación de vuelo individual	26
2.20. Desplazamiento en el plano XZ para simulación de vuelo individual	27
2.21. Desplazamiento en x para simulación de vuelo simultaneo	28
2.22. Desplazamiento en z para simulación de vuelo simultaneo	28
2.23. Desplazamiento en el plano XZ para simulación de vuelo simultaneo	29
3.1. Diagrama de la plataforma experimental	32
3.2. Cámaras infrarrojas	34
3.3. Distribución de marcadores infrarrojos sobre un AR.Drone 2.0	34
3.4. Distribución de las cámaras	35
3.5. Trayectoria del cuadrirotor en x	37
3.6. Trayectoria del cuadrirotor en y	38

3.7. Trayectoria del cuadrirotor en z	38
3.8. Trayectoria del cuadrirotor en ψ	39
3.9. Trayectoria del cuadrirotor en el plano XZ	39
3.10. Trayectoria del cuadrirotor 1 en x	40
3.11. Trayectoria del cuadrirotor 1 en y	41
3.12. Trayectoria del cuadrirotor 1 en z	41
3.13. Trayectoria del cuadrirotor 1 en ψ	42
3.14. Trayectoria del cuadrirotor 2 en x	42
3.15. Trayectoria del cuadrirotor 2 en y	43
3.16. Trayectoria del cuadrirotor 2 en z	43
3.17. Trayectoria del cuadrirotor 2 en ψ	44
3.18. Trayectorias líder-seguidor en x	45
3.19. Trayectorias líder-seguidor en y	45
3.20. Trayectorias líder-seguidor en z	46
3.21. Trayectorias líder-seguidor en ψ	46
A.1. Bloques empleados para la simulación	56
A.2. Otros bloques empleados para la simulación	58
A.3. Subsistema representando a un cuadrirotor	58
A.4. Diagrama a bloques empleado para simulación de vuelo individual . .	59
A.5. Diagrama a bloques empleado para simulación de vuelo simultaneo .	60

Lista de tablas

2.1. Estados medibles y estados requeridos	20
------------------------------------------------------	----

Capítulo 1

Introducción

En este capítulo se introducen los sistemas multi-agente y sus posibles áreas de aplicación. Se presenta una breve introducción a los vehículos aéreos no tripulados y, de forma más específica, al cuadricóptero, el cual es empleado como parte de la plataforma experimental en el presente trabajo. De igual forma, se plantean los objetivos a alcanzar y se describen las aportaciones de la presente tesis. Por último, se explica la organización de la misma.

1.1. Sistemas multi-agente.

De forma general un agente es un ente (persona, animal o dispositivo) que realiza una acción [1]. Un sistema multi-agente (*Multi-Agent System*, MAS) es un sistema compuesto por múltiples agentes que interactúan entre sí, con el objeto de resolver problemas que son difíciles o imposibles de resolver para un agente de forma individual. En un sistema multi-agente el conjunto de agentes está sometido continuamente a cambios locales. Estos cambios se diseñan mediante reglas de comportamiento, cuyos resultados están influenciados por el comportamiento del resto de agentes. Un sistema multi-agente es un sistema que reúne los siguientes elementos:

- Un entorno
- Un conjunto de agentes que representan las entidades activas del sistema.

- Un conjunto de relaciones que unen agentes.

El estudio de los sistemas multi-agente está orientado al desarrollo y análisis de resoluciones a problemas sofisticados de inteligencia artificial y arquitecturas de control tanto para sistemas con un único agente como sistemas multi-agente. Los temas de investigación en sistemas multi-agente son:

- ingeniería de software orientado a agentes
- cooperación y coordinación
- organización
- comunicación
- resolución distribuida de problemas
- aprendizaje de multi-agentes
- fiabilidad y tolerancia a fallos

1.2. Vehículos aéreos no tripulados.

Un vehículo aéreo no tripulado (*Unmanned Aerial Vehicle*, UAV), es una aeronave capaz de volar de manera autónoma. El control cooperativo de un grupo de UAVs resulta de gran interés en aplicaciones de búsqueda y rescate, vigilancia e inspección a distancia, por mencionar algunas. Éstas tareas pueden resultar repetitivas o peligrosas, haciéndolas ideales para que sean realizadas por los UAV's.

Una de las aeronaves no tripuladas que ha llamado la atención en las últimas décadas, es la que es capaz de realizar un despegue y aterrizaje vertical (Vertical Take-Off and Landing, VTOL), ya que una vez realizado el despegue, el VTOL tiene posibilidades de realizar movimientos en cualquier dirección del espacio tridimensional. En particular, cuando la aeronave VTOL realiza un despegue vertical y movimientos sobre un plano vertical, se le denomina PVTOL (Planar Vertical Take Off and Landing).

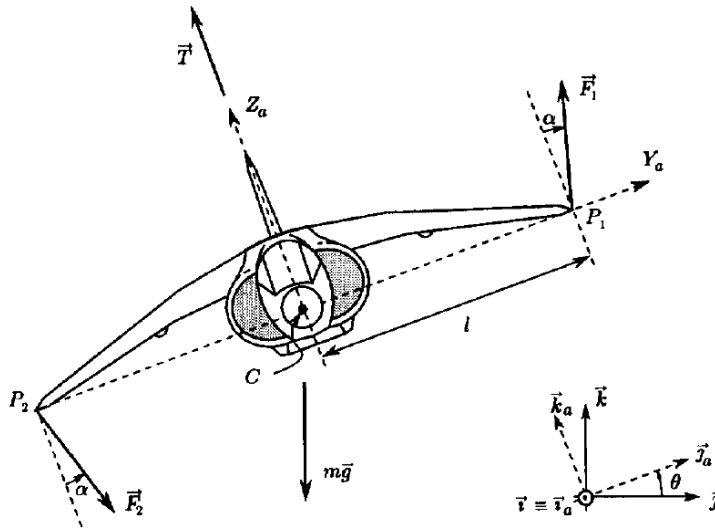


Figura 1.1: Modelo de aeronave Harrier

Algunos de los principales problemas relacionados con el PVTOL que la comunidad científica y tecnológica ha enfrentado, son la estabilización y el seguimiento de trayectorias. En [4] se estudió la aeronave tripulada Harrier (Fig. 1.1), esta aeronave fue concebida para poder efectuar despegue y aterrizaje vertical, ya que es común que se requiera despegar desde una embarcación la cual no cuenta con una pista lo suficientemente larga como para efectuar despegue y aterrizaje de aviones convencionales.

Implementar en tiempo real alguna ley de control para estabilizar al VTOL y/o hacer que siga alguna trayectoria, nos conduce a otro importante reto: la medición de todos sus estados. La implementación de cualquier ley de control requiere del conocimiento de todas las coordenadas y velocidades de la aeronave. Un importante reto para controlarlo en tiempo-real es la medición de los ángulos de rotación y las correspondientes velocidades angulares. La dificultad reside en que los ángulos pueden ser muy pequeños para producir movimientos horizontales, y por ello se requieren de instrumentos de precisión y de alta resolución.



Figura 1.2: Operación del cuadrirotor

1.3. Estado del arte.

El cuadrirotor es una aeronave de despegue y aterrizaje vertical [5] que consiste de 4 hélices dispuestas en forma de cruz. Un cuadrirotor consta de dos hélices girando en sentido horario y dos hélices más girando en sentido antihorario que se oponen diagonalmente entre sí como se muestra en la Fig. 1.2. Esto da como resultado que la fuerza de reacción de cada hélice se cancele por el componente reactivo de la hélice diagonalmente opuesta. Esto elimina la necesidad de un rotor de cola tipo helicóptero. La simetría en el cuerpo del cuadrirotor da simplicidad en el diseño del controlador, ya que puede ser controlado a través de la variación de la velocidad de las cuatro hélices [6].

Las naves cuadrirotor son eficientes y presentan ventajas mecánicas, ya que el tener cuatro hélices pequeñas en lugar de una sola hélice grande reduce el par torsor en el sistema lo que significa que las hélices pueden ser accionadas a velocidades más altas sin producir vibraciones mecánicas adicionales lo que también aumenta la eficiencia del motor. Un menor par y vibraciones someten a una menor tensión a los componentes mecánicos de la aeronave.

Los cuadrirotos aceptan una mayor carga útil y gran variedad de maniobras comparado con otros tipos de aeronaves; permiten un despegue y aterrizaje vertical así como volar en zonas de difícil acceso [7], pudiendo acceder a zonas en las que no se puede acceder con aviones ni helicópteros tradicionales [8]. Hoy en día, los cuadrirotos pueden realizar maniobras rápidas y complejas [9], navegar de forma autónoma tanto en ambientes conocidos [10] como desconocidos [11] y cooperar en tareas de manipulación y transporte de objetos [12].

En la literatura se pueden encontrar diferentes algoritmos de control para los problemas de regulación y de seguimiento de la trayectoria para un solo cuadrirotor. Por ejemplo, linealización por retroalimentación [13], [14], retroalimentación de la salida basado en la teoría de Lyapunov [15], backstepping [16], saturaciones anidadas [17], control por modos deslizantes [18] y control H_∞ no lineal [19]. También se han propuesto algunas soluciones para coordinar un sistema multi-agente compuesto de varios cuadrirotos. En [20] se propone un esquema de control basado en la teoría de campos potenciales para que un grupo de cuadrirotos logren realizar una formación deseada con la restricción de que la altura de vuelo sea constante. En [21] se presenta la implementación en tiempo real basada en el modelo de Giotto para coordinar un sistema multi-agente compuesto de dos cuadrirotos utilizando el esquema líder-seguidor. Los cuadrirotos guardan una posición relativa constante entre ellos mientras realizan el seguimiento de una trayectoria deseada. Por otra parte, el trabajo [22] considera el problema de controlar vehículos aéreos que se mueven en un ambiente tridimensional mientras mantienen una formación cerrada la cual se especifica a través de una matriz que define la separación relativa entre los robots. En [23] se presenta una estrategia para coordinar a un líder virtual y a un seguidor. Esta implementación combina un controlador de modos deslizantes para la dinámica traslacional con un controlador proporcional-derivativo para la dinámica rotacional.

El presente trabajo presenta un esquema de control basado en funciones de transferencia obtenidas a partir de la respuesta a la señal pulso unitario para que un sistema

multi-agente, conformado por dos cuadrirotos, realice el seguimiento de una trayectoria utilizando el esquema de líder-seguidor.

1.4. Objetivos de la tesis.

1.4.1. Objetivo general.

Realizar investigación teórica y experimental enfocada a las aeronaves VTOL tipo cuadrirotor, de forma mas especifica al AR.DRONE 2.0, con el propósito de realizar seguimiento de trayectorias predefinidas en el espacio tridimensional, en configuración líder-seguidor.

1.4.2. Objetivos específicos.

1. Obtención del modelo tipo caja negra, en tiempo-continuo, del cuadrirotor.
2. Diseño de un controlador para efectuar el seguimiento de trayectorias aplicable a una aeronave comercial AR.DRONE 2.0.
3. Implementación del controlador para efectuar seguimiento de trayectorias en configuración líder-seguidor en un sistema multi-agente.
4. Validación del controlador mediante simulaciones numéricas.
5. Validación del controlador mediante experimentos en tiempo-real.

1.5. Aportaciones de la tesis.

Las aportaciones del presente trabajo son:

- Estimación de las funciones de transferencia que describen el comportamiento dinámico del AR.DRONE 2.0 para cada una de las variables de control.
- Diseño de las estrategias de control necesarias para llevar a cabo el seguimiento de trayectorias dentro del sistema multi-agente, manteniendo la formación geométrica requerida.
- Validación de dichas estrategias de control en una plataforma experimental, en tiempo real.

1.6. Organización de la tesis.

El trabajo está organizado de la siguiente manera: El Capítulo 2 presenta el planteamiento del problema, la identificación de las funciones de transferencia, el desarrollo de las leyes de control que permiten a los cuadrirotos realizar el seguimiento de trayectorias utilizando el esquema líder-seguidor, así como los resultados obtenidos a partir de una serie de simulaciones computacionales. La descripción de la plataforma, así como los resultados experimentales se presentan en el Capítulo 3. Por último, en el Capítulo 4, se presentan algunas conclusiones y trabajo futuro.

Capítulo 2

Planteamiento del problema

En este capítulo se plantea el problema a resolver. Se aborda el método con el que se obtiene el modelo del cuadrirotor. Posteriormente se presenta el problema de seguimiento de trayectorias en configuración líder-seguidor. Se explica el método utilizado para la estimación de la velocidad partiendo de posiciones a través de una derivada numérica. Finalmente se presenta el controlador implementado en ambos cuadrirotos así como las gráficas obtenidas a partir de una serie de simulaciones numéricas.

2.1. Modelo dinámico del Cuadrirotor

Siendo el cuadrirotor un objeto capaz de volar a través del espacio, son necesarias una serie de coordenadas para describir su posición y orientación tomando como referencia al sistema inercial $X_0Y_0Z_0$ (Fig. 2.1). Las coordenadas generalizadas del cuadrirotor están dadas por la posición del centro de masa dada por el vector $\xi = [x, y, z]^T$ y la orientación dada por el vector $\Omega = [\theta, \phi, \psi]^T$ conformado por los ángulos de Euler. Los ángulos de Euler constituyen un conjunto de tres coordenadas angulares que sirven para especificar la orientación de un sistema de referencia de ejes ortogonales, normalmente móvil, respecto a otro sistema de referencia de ejes ortogonales normalmente fijos. Esto da como resultado el vector

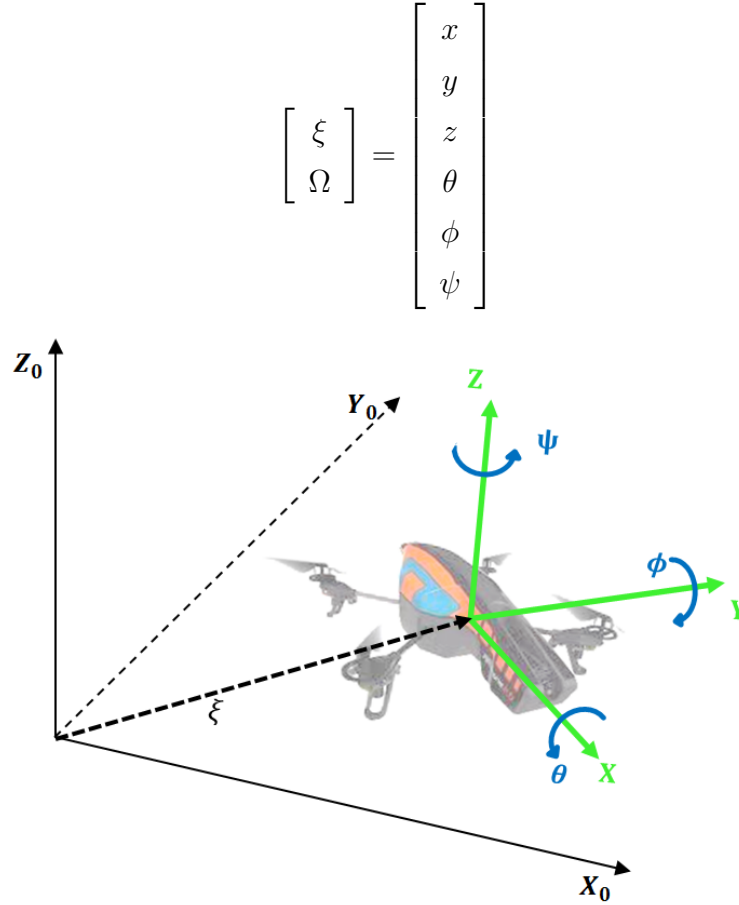


Figura 2.1: Diagrama de un AR.Drone 2.0 en el espacio

Basándonos en [17], usamos el siguiente modelo dinámico:

$$m\ddot{x} = -u (\cos(\psi) \sin(\theta) \cos(\phi) + \sin(\psi) \sin(\phi))$$

$$m\ddot{y} = -u (\sin(\psi) \sin(\theta) \cos(\phi) - \cos(\psi) \sin(\phi))$$

$$m\ddot{z} = -u (\cos(\theta) \cos(\phi)) + mg$$

$$\ddot{\psi} = \tau_{\psi}$$

$$\ddot{\theta} = \tau_{\theta}$$

$$\ddot{\phi} = \tau_{\phi}$$

donde u es la fuerza total de empuje generada en conjunto por cada una de las hélices, m representa la masa del cuadricóptero, g denota la constante gravitacional, τ_{ψ} , τ_{θ} y τ_{ϕ}

son los torques de guiñada, alabeo y cabeceo respectivamente.

2.2. Modelo del AR.Drone 2.0

Para llevar a cabo las diferentes maniobras del AR.Drone 2.0 a través del espacio es necesario modificar la velocidad a la que las 4 hélices giran, al llevar a cabo estas variaciones es posible modificar sus ángulos de cabeceo θ , alabeo ϕ y guiñada ψ , así como modificar la altura z a la que éste se encuentra.

El AR.Drone 2.0 cuenta con un controlador de navegación embebido que se encarga de regular las velocidades a las que cada hélice debe girar para lograr las maniobras deseadas [24]. Dicho control de navegación permite solamente la entrada de cuatro variables de control, las cuales son:

- La velocidad de ascenso/descenso (u_z).
- La velocidad de guiñada (u_ψ).
- El ángulo de alabeo (u_θ).
- El ángulo de cabeceo (u_ϕ).

Debido a la naturaleza hermética del controlador embebido, el usuario no tiene acceso a todos los estados del sistema, por lo que se propone modelar al AR.Drone como un conjunto de sistemas lineales (Fig. 2.2) 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.

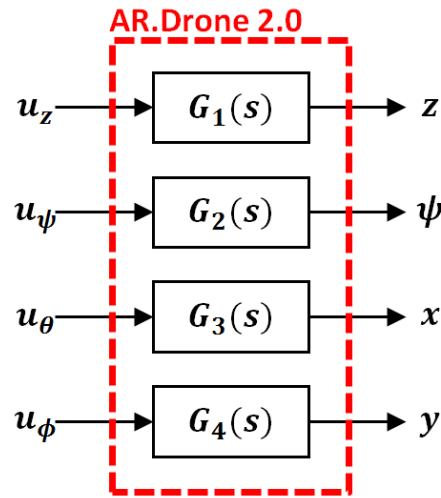


Figura 2.2: Diagrama entradas / salidas del AR.Drone 2.0

Sabemos de [24] que la dinámica del control angular del AR.Drone 2.0 es de segundo orden, por lo que el diagrama presentado en la Fig. 2.2 puede ser desglosado como sigue (Fig. 2.3)

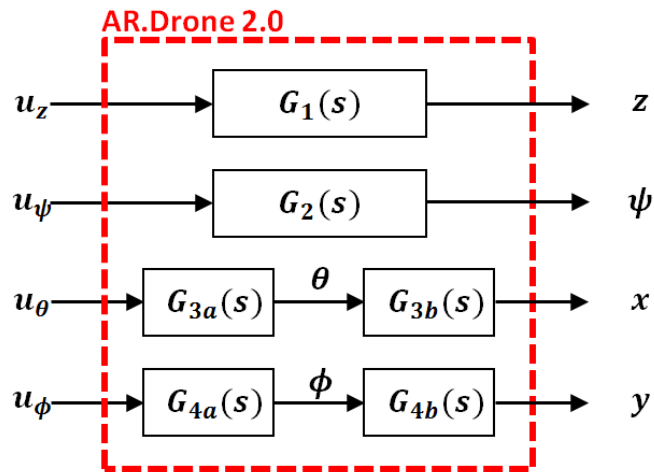


Figura 2.3: Diagrama entradas / salidas extendido del AR.Drone 2.0

Siendo $G_1(s)$, $G_2(s)$, $G_{3a}(s)$ y $G_{4a}(s)$ funciones de transferencia de segundo orden. Para determinar el orden de $G_{3b}(s)$ y $G_{4b}(s)$ se plantea lo siguiente.

Del diagrama de cuerpo libre (Fig. 2.4) del cuadricóptero en un punto de equilibrio en

el plano XZ podemos deducir que

$$F = mg \quad (2.1)$$

siendo:

F : fuerza de sustentación

m : masa del cuadrirotor

g : aceleración de la gravedad

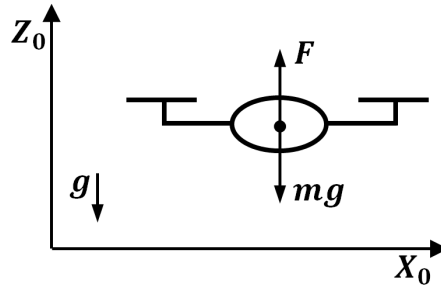


Figura 2.4: Diagrama de cuerpo libre en estado de flotación

Para generar algún desplazamiento sobre X es necesario producir un cambio en θ

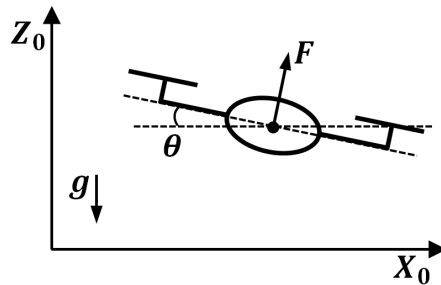


Figura 2.5: Diagrama de cuerpo libre en estado de flotación con θ modificado

Del diagrama de cuerpo libre de la Fig. 2.5 obtenemos

$$m\ddot{x} = F \sin(\theta) \quad (2.2)$$

Para ángulos pequeños, es posible realizar la aproximación $\sin(\theta) \approx \theta$. Entonces

$$m\ddot{x} = F\theta \quad (2.3)$$

Sustituyendo (2.1) en (2.3) y reduciendo obtenemos

$$\ddot{x} = g\theta \quad (2.4)$$

Aplicando transformada de Laplace a (2.4)

$$X(s)s^2 = g\Theta(s) \quad (2.5)$$

Despejando (2.5)

$$\frac{X(s)}{\Theta(s)} = \frac{g}{s^2} \quad (2.6)$$

De (2.6) podemos observar la estructura deseada para la función de transferencia

$$G_{3b}(s) = \frac{X(s)}{\Theta(s)} = \frac{g}{s^2} \quad (2.7)$$

siendo esta un doble integrador. De manera similar es posible determinar que

$$G_{4b}(s) = \frac{Y(s)}{\Phi(s)} = \frac{g}{s^2} \quad (2.8)$$

Para determinar las funciones de transferencia buscadas asumimos como hipótesis de trabajo que las 4 variables de control (u_z , u_ψ , u_θ y u_ϕ) están desacopladas (no tienen interacción entre si). Dicha hipótesis se valida de manera experimental introduciendo un valor sobre alguna de las variables de control con el fin de producir un desplazamiento sobre su respectivo estado. Los estados afectados por las entradas de control u_z , u_ψ , u_θ y u_ϕ son z , ψ , x e y respectivamente. Se busca que el resto de los estados relacionados a otras entradas de control no se vean afectados o que los cambios registrados sean despreciables.

La Fig. 2.6 muestra los desplazamientos registrados para una entrada $u_z = 0.1$. Los desplazamientos registrados son $\Delta x = 0.03m$, $\Delta y = 0.03m$, $\Delta z = 1.06m$ y $\Delta\psi = 3.7^\circ$.

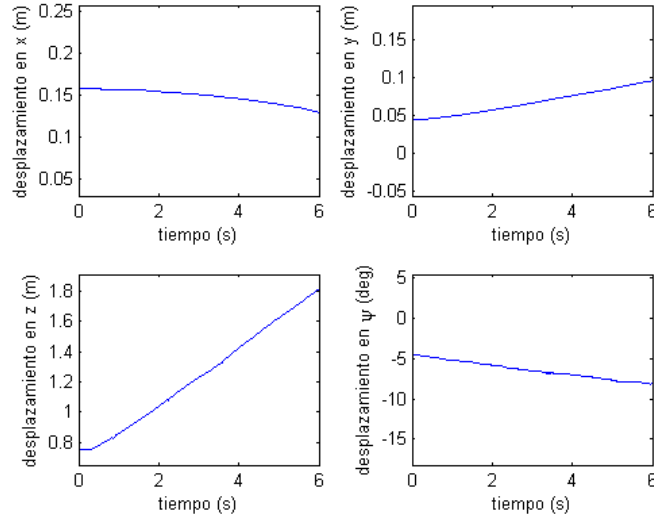


Figura 2.6: Desplazamientos registrados para una entrada u_z constante

La Fig. 2.7 muestra los desplazamientos registrados para una entrada $u_\psi = 0.1$. Los desplazamientos registrados son $\Delta x = 0.03m$, $\Delta y = 0.06m$, $\Delta z = 0.02m$ y $\Delta\psi = 59.5^\circ$.

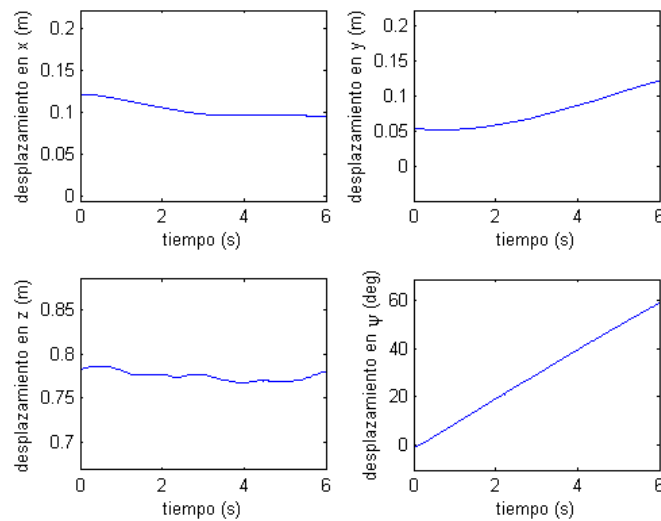


Figura 2.7: Desplazamientos registrados para una entrada u_ψ constante

La Fig. 2.8 muestra los desplazamientos registrados para una entrada $u_x = 0.1$. Los desplazamientos registrados son $\Delta x = 2.57m$, $\Delta y = 0.04m$, $\Delta z = 0.10m$ y $\Delta\psi = 2.96^\circ$.

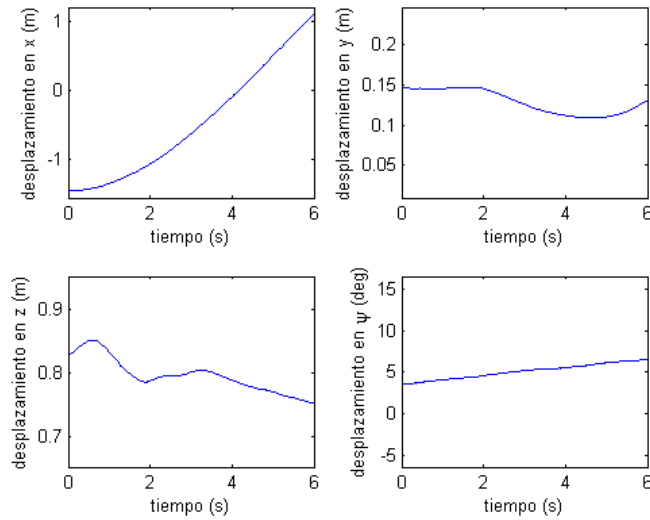


Figura 2.8: Desplazamientos registrados para una entrada u_θ constante

La Fig. 2.9 muestra los desplazamientos registrados para una entrada $u_y = 0.1$. Los desplazamientos registrados son $\Delta x = 0.05m$, $\Delta y = 2.16m$, $\Delta z = 0.06m$ y $\Delta\psi = 2.59^\circ$.

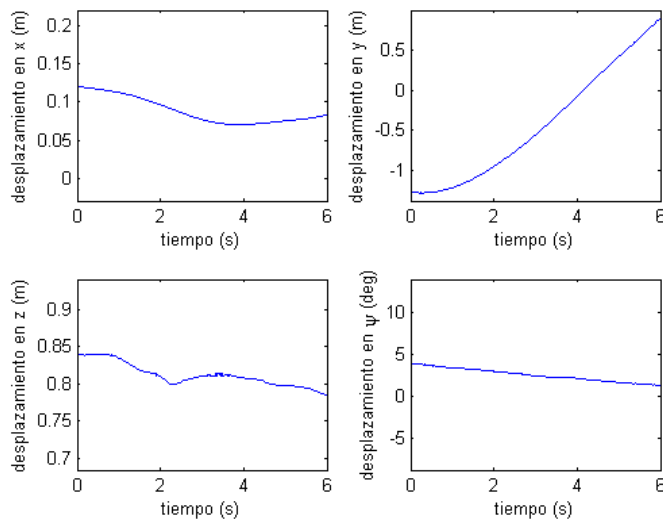


Figura 2.9: Desplazamientos registrados para una entrada u_ϕ constante

Es posible observar que el mayor desplazamiento se registra sobre el estado esperado siendo el resto de ellos prácticamente despreciables, por lo que se supone la hipótesis propuesta como correcta.

Se propone la obtención de las funciones de transferencia requeridas partiendo de la respuesta a una señal tipo pulso unitario (Fig. 2.10) definida como

$$\delta(t) = \begin{cases} 0.1 & \text{si } 0 \leq t \leq 1 \\ 0 & \text{si } t < 0 \text{ ó } t > 1 \end{cases}$$

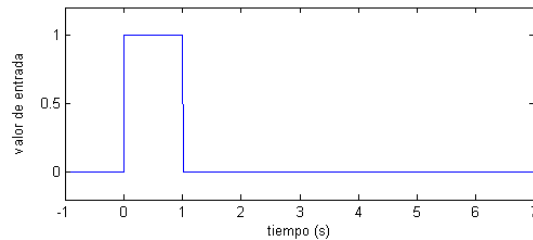


Figura 2.10: Pulso unitario

Las respuestas obtenidas para la entrada tipo pulso unitario son las presentadas en la Fig. 2.11

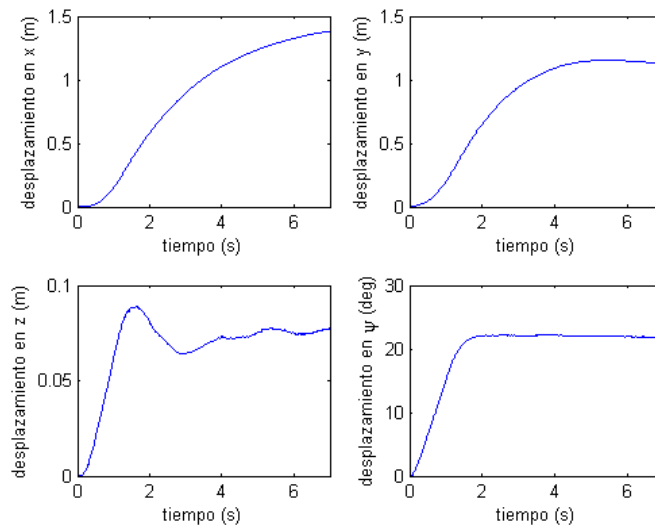


Figura 2.11: Respuestas a entrada pulso

Una vez obtenidos estos datos, se utilizó MATLAB para identificar las funciones de transferencia por medio de una aproximación por mínimos cuadrados. Las funciones de transferencia obtenidas son:

$$\begin{aligned}
 G_1(s) &= \frac{Z(s)}{U_z(s)} = \frac{0.5}{s^2 + 4s} \\
 G_2(s) &= \frac{\Psi(s)}{U_\psi(s)} = \frac{57}{s^2 + 3s} \\
 G_3(s) &= \frac{\Theta(s) X(s)}{U_\theta(s) \Theta(s)} = \frac{430}{s^4 + 17s^3 + 75s^2 + 27s} \\
 G_4(s) &= \frac{\Phi(s) Y(s)}{U_\phi(s) \Phi(s)} = \frac{120}{s^4 + 16s^3 + 38s^2 + 17s}
 \end{aligned}$$

Siendo estas las funciones de transferencia que describen la dinámica interna al AR.Drone 2.0. Es posible observar que el número de estados contenidos en las funciones de transferencia es igual 12, mismos que son expresados en las ecuaciones que describen al modelo dinámico del cuadrirotor.

2.3. Formación orientada al líder.

El objetivo de control es lograr una formación entre dos agentes. Un líder R_1 , el cual se encarga de seguir una trayectoria predefinida $m(t)$ y un seguidor R_2 el cual se encuentra en una cierta posición relativa del líder definida por el vector $c_{12} \in \mathbb{R}^3$ (Fig 2.12). Tanto el líder R_1 como el seguidor R_2 son aeronaves tipo cuadrirotor AR.Drone 2.0.

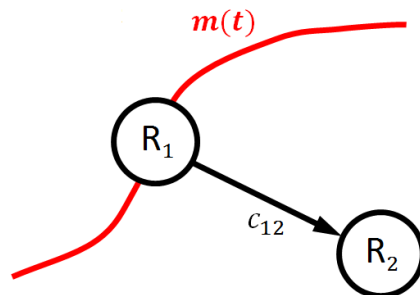


Figura 2.12: Formación orientada al líder

Sea $N = \{R_1, R_2\}$ un conjunto de agentes moviéndose a través del espacio XYZ con posiciones $\alpha_i = (p_i, q_i, r_i)^T$, $i = 1, 2$, el agente R_1 es el líder del sistema y debe seguir asintóticamente una trayectoria predeterminada. El otro agente R_2 actúa como seguidor y debe mantener una posición deseada con respecto al líder. Sea $m(t) = [m_x(t), m_y(t), m_z(t)]^T$ la trayectoria predeterminada, la cual se supone continua y suficientemente diferenciable. En este trabajo se utiliza una gráfica de formación directa de cadena abierta con un líder, también conocida como configuración convoy [25], [26]. Ahora definimos α_i^* como la posición deseada del agente R_i , dentro de la formación deseada. En particular, en este trabajo establecemos

$$\begin{aligned}\alpha_2^* &= \alpha_1 + c_{12} \\ \alpha_1^* &= m(t)\end{aligned}\tag{2.9}$$

donde $c_{12} = [c_{12x}, c_{12y}, c_{12z}]^T \in \mathbb{R}^3$ es el vector de formación relativo del agente R_2 con respecto al agente R_1 . Asumimos que el vector de formación c_{12} es constante.

El objetivo de control es diseñar una ley de control tal que

$$\begin{aligned}\lim_{t \rightarrow \infty} (\alpha_1(t) - m(t)) &= 0 \\ \lim_{t \rightarrow \infty} (\alpha_2(t) - \alpha_2^*(t)) &= 0\end{aligned}\tag{2.10}$$

2.4. Estimación de la velocidad.

El AR.DRONE 2.0 cuenta con múltiples sensores a bordo, capaces de proporcionar mediciones de algunos de los estados que serán de utilidad para la implementación de las leyes de control presentadas más adelante. Desafortunadamente, no es posible recibir la información de todos los estados necesarios desde el AR.DRONE 2.0, por lo que será necesario recurrir a un sistema externo de cámaras el cual se describe en el Capítulo 3. En la tabla 2.1 se muestra un listado de todos los estados, desde donde pueden ser medidos y si estos son requeridos.

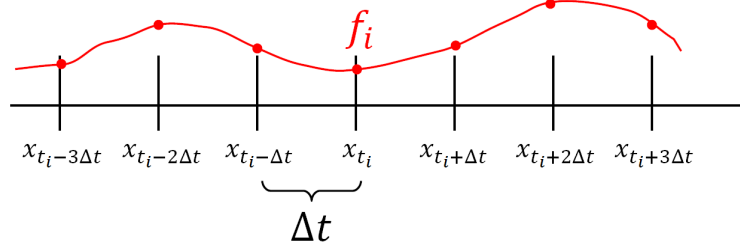
Todos los estados requeridos pueden ser medidos de alguna fuente, a excepción de la velocidad de ascenso/descenso \dot{z} , por lo que proponemos una derivación numérica

Fuente / Estado	x	y	z	θ	ϕ	ψ	\dot{x}	\dot{y}	\dot{z}	$\dot{\theta}$	$\dot{\phi}$	$\dot{\psi}$
AR.DRONE 2.0	×	×	✓	×	×	×	✓	✓	×	×	×	×
Sistema de visión	✓	✓	✓	✓	✓	✓	×	×	×	×	×	×
Requerido	✓	✓	✓	×	×	✓	✓	✓	✓	×	×	×

Tabla 2.1: Estados medibles y estados requeridos

para estimarla.

Sea $f(t)$ una función diferenciable definida por una serie finita de puntos $\{x_0, x_1, \dots, x_i, \dots\}$ separados por una distancia constante Δt entre sí.



Los valores que la función $f(t)$ toma en los puntos x_{t_i} son

$$f_i = f(x_{t_i}) \quad (2.11)$$

Evaluamos (2.11) en $x_{t_i+\Delta t}$

$$f(x_{t_i+\Delta t}) = f_{i+1} \quad (2.12)$$

Tomemos ahora la expansión de $f(t)$ en serie de Taylor alrededor del punto x_{t_i}

$$f_{i+1} = f_i + \left. \frac{\partial f}{\partial t} \right|_{x_{t_i}} \Delta t + \frac{1}{2} \left. \frac{\partial^2 f}{\partial t^2} \right|_{x_{t_i}} (\Delta t)^2 + \dots \quad (2.13)$$

Despejando (2.13)

$$\left. \frac{\partial f}{\partial t} \right|_{x_{t_i}} = \frac{f_{i+1} - f_i}{\Delta t} - \frac{1}{2} \left. \frac{\partial^2 f}{\partial t^2} \right|_{x_{t_i}} \Delta t + \dots \quad (2.14)$$

Si despreciamos los términos de orden superior en (2.14), obtenemos una aproximación a la derivada con dos puntos

$$f'(x_{t_i}) \approx \frac{f_{i+1} - f_i}{\Delta t} \quad (2.15)$$

Siguiendo el mismo método evaluando (2.11) en f_{i-1} con una separación de $2\Delta t$ obtenemos una aproximación a la derivada centrada entre dos puntos

$$f'(x_{t_i}) \approx \frac{f_{i+1} - f_{i-1}}{2\Delta t} \quad (2.16)$$

Para validar el uso de (2.15) y (2.16) como aproximaciones a la derivada de una función conformada utilizando una serie de mediciones, se emplea MATLAB para generar la función $f(x) = \sin(\omega t)$ (Fig. 2.13) a la que se le adiciona un número aleatorio para simular la presencia de ruido en la medición.

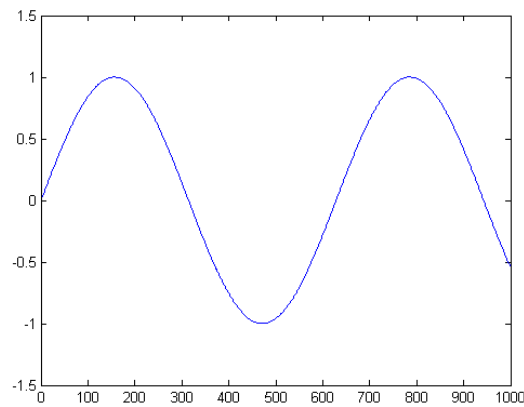
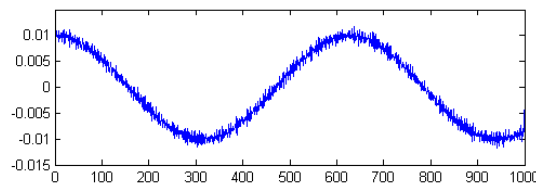
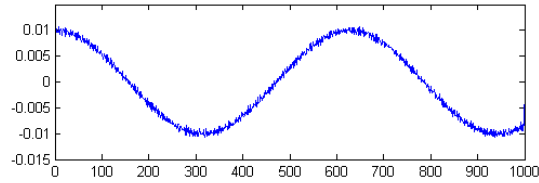


Figura 2.13: Señal de prueba

Al implementar (2.15) y (2.16) a la función descrita, obtenemos las Fig. 2.14(a) y 2.14(b) respectivamente.



(a) Estimación de la derivada basada en (2.15)



(b) Estimación de la derivada basada en (2.16)

Figura 2.14: Derivadas estimadas

De forma analítica sabemos que $f'(x) = \omega \cos(\omega * t)$, por lo que es posible observar que el resultado es el esperado.

Al contar con ruido la señal original, este es amplificado al aplicar la derivada. Siendo menos evidente este efecto en la derivada estimada a partir de (2.16), por lo que se decide su uso para la estimación de las derivadas en el presente trabajo.

2.5. Lazos de control.

La dinámica del subsistema de guiñada tiene una respuesta muy rápida sin la presencia de sobreimpulso, por lo tanto un controlador proporcional es suficiente. La dinámica de la altitud es un poco más lenta con la presencia de sobreimpulso por lo que se implementa un controlador PD para fines de estabilización altitud. Es posible observar que la dinámica de X se trata de un subsistema sobreamortiguado, por lo que es requerido un controlador PD para lograr el seguimiento de trayectorias a velocidades altas. El mismo razonamiento se emplea para la dinámica de Y . La Figura 2.15 presenta el diagrama a bloques para el control de las 4 variables desacopladas.

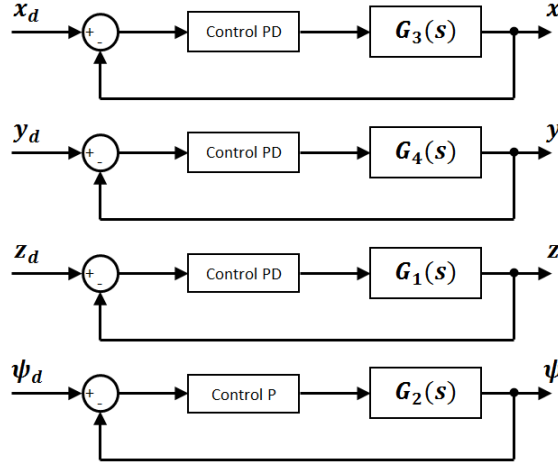


Figura 2.15: Diagrama de control

Las ecuaciones que describen dichos lazos de control están dadas por

$$\begin{aligned}
 u_z &= -K_{pz} (z - z_d) - K_{dz} (\dot{z} - \dot{z}_d) \\
 u_\psi &= -K_{p\psi} (\psi - \psi_d) \\
 u_\theta &= -K_{p\theta} (x - x_d) - K_{d\theta} (\dot{x} - \dot{x}_d) \\
 u_\phi &= -K_{p\phi} (y - y_d) - K_{d\phi} (\dot{y} - \dot{y}_d)
 \end{aligned} \tag{2.17}$$

donde K_{pz} , $K_{p\psi}$, $K_{p\theta}$, $K_{p\phi} > 0$ corresponden a las ganancias proporcionales y K_{dz} , $K_{d\theta}$, $K_{d\phi} > 0$ corresponden a las ganancias derivativas.

2.5.1. Ley de control para el agente líder.

Las ganancias proporcionales están dadas por K_{pz} , $K_{p\psi}$, $K_{p\theta}$, $K_{p\phi} > 0$ y las ganancias derivativas por K_{dz} , $K_{d\theta}$, $K_{d\phi} > 0$. La posición deseada para el agente líder a través del espacio es la descrita por la trayectoria $m(t) = [m_x(t), m_y(t), m_z(t)]^T$, así como la orientación deseada la descrita por el vector $m_{\psi 1}(t)$.

Con base en lo anterior la ley de control para el agente líder es:

$$\begin{aligned}
 u_{z1} &= -K_{pz}(z_1 - m_z(t)) - K_{dz}(\dot{z}_1 - \dot{m}_z(t)) \\
 u_{\psi1} &= -K_{p\psi}(\psi_1 - m_{\psi1}(t)) \\
 u_{\theta1} &= -K_{p\theta}(x_1 - m_x(t)) - K_{d\theta}(\dot{x}_1 - \dot{m}_x(t)) \\
 u_{\phi1} &= -K_{p\phi}(y_1 - m_y(t)) - K_{d\phi}(\dot{y}_1 - \dot{m}_y(t))
 \end{aligned} \tag{2.18}$$

2.5.2. Ley de control para el agente seguidor.

La ley de control del seguidor depende de la posición y velocidad del líder, así como del vector de posición $c_{12} = [c_{12x}, c_{12y}, c_{12z}]^T$. Quedando los parámetros de la trayectoria deseada para el agente seguidor como: $x_{d2} = x_1 + c_{12x}$, $y_{d2} = y_1 + c_{12y}$, $z_{d2} = z_1 + c_{12z}$ así como la orientación deseada la descrita por el vector $m_{\psi2}(t)$.

La ley de control para el agente seguidor es:

$$\begin{aligned}
 u_{z2} &= -K_{pz}(z_2 - (z_1 + c_{12z})) - K_{dz}(\dot{z}_2 - \dot{z}_1) \\
 u_{\psi2} &= -K_{p\psi}(\psi_2 - m_{\psi2}(t)) \\
 u_{\theta2} &= -K_{p\theta}(x_2 - (x_1 + c_{12x})) - K_{d\theta}(\dot{x}_2 - \dot{x}_1) \\
 u_{\phi2} &= -K_{p\phi}(y_2 - (y_1 + c_{12y})) - K_{d\phi}(\dot{y}_2 - \dot{y}_1)
 \end{aligned} \tag{2.19}$$

2.6. Simulaciones numéricas

En esta sección se presentan una serie de simulaciones numéricas basadas en el modelo del AR.DRONE 2.0 previamente obtenido a partir de las funciones de transferencia así como las leyes de control previamente propuestas. Dichas simulaciones son realizadas con la ayuda de MATLAB en el entorno de programación visual SIMULINK. Los bloques usados se presentan en el apéndice A, tanto el referente a las aeronaves como el empleado para implementar el controlador. De igual forma se presentan simulaciones del vuelo de una sola aeronave así como del vuelo simultaneo de ambas.

2.7. Simulación de vuelo individual.

Se muestran las gráficas resultantes de la presente simulación en las Fig. 2.16, 2.17, 2.18, 2.19 y 2.20, siendo la trayectoria deseada para el plano XZ una lemniscata para

$m_y = 0$ y $m_\psi = 0$ bajo condiciones iniciales arbitrarias.

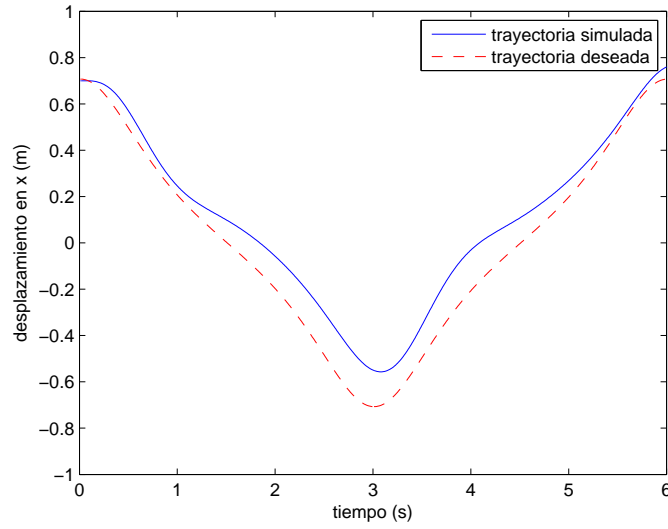


Figura 2.16: Desplazamiento en x para simulación de vuelo individual

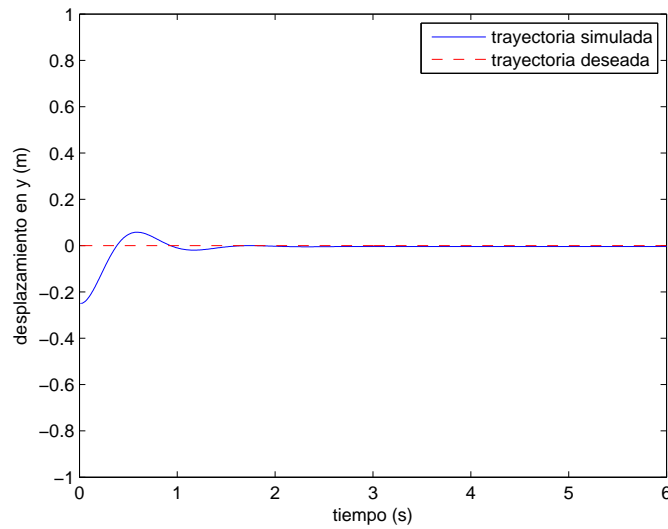
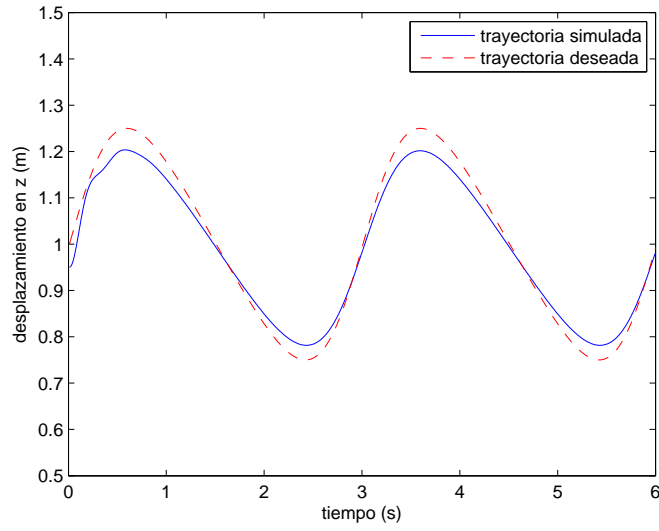
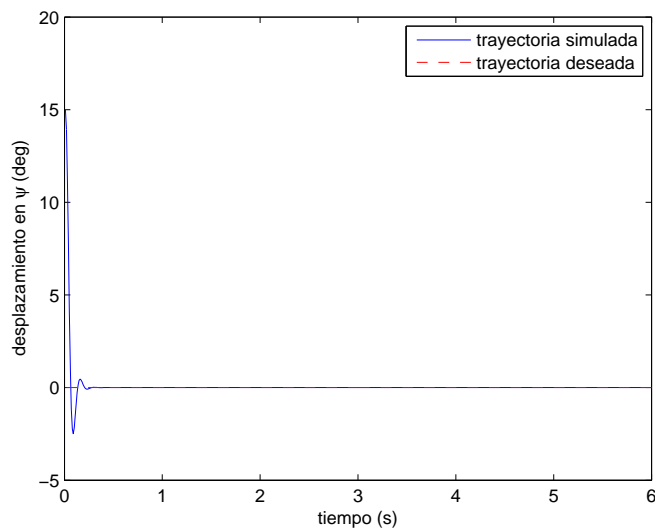


Figura 2.17: Desplazamiento en y para simulación de vuelo individual

Figura 2.18: Desplazamiento en z para simulación de vuelo individualFigura 2.19: Desplazamiento en ψ para simulación de vuelo individual

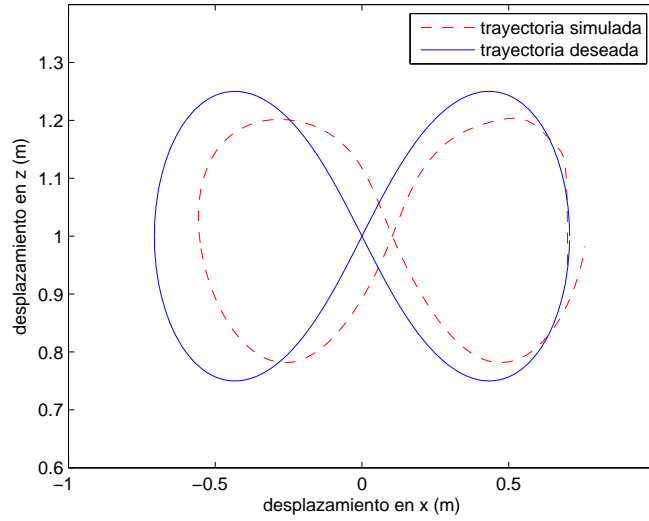


Figura 2.20: Desplazamiento en el plano XZ para simulación de vuelo individual

Ya que el seguimiento de trayectorias se lleva de manera apropiada, se procede a la simulación de vuelo simultaneo en configuración líder-seguidor.

2.8. Simulación de vuelo simultaneo en configuración líder-seguidor

Una vez validadas las leyes de control mediante la simulación previa, se procede a la simulación de vuelo simultaneo. Se emplean para el agente seguidor las leyes de control descritas en (2.19), con la siguiente configuración:

$$c_{12x} = 0$$

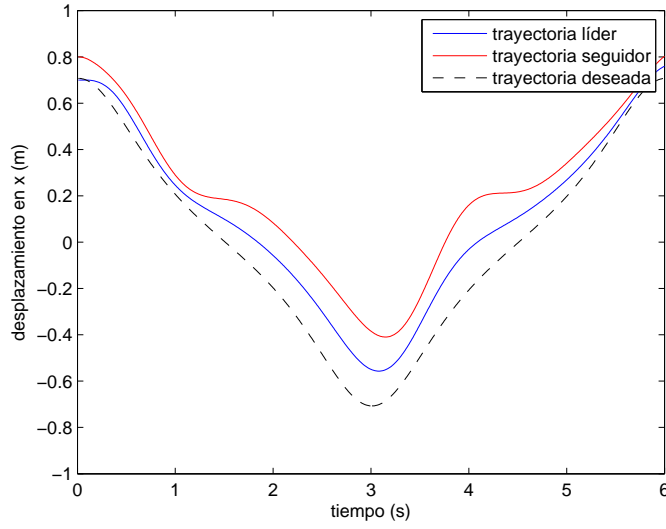
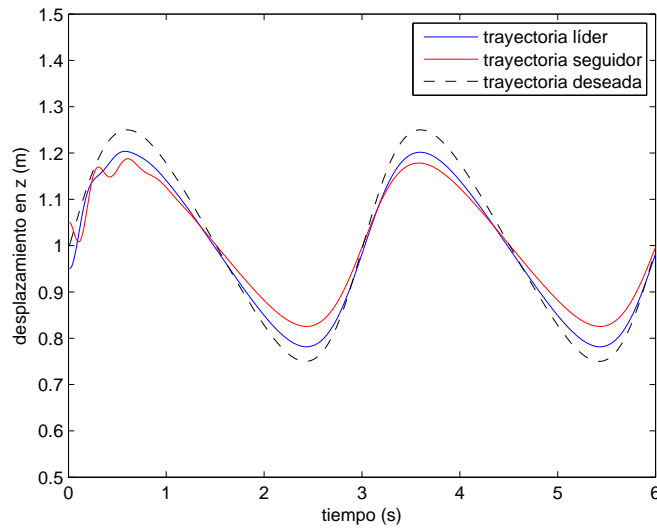
$$c_{12y} = 2$$

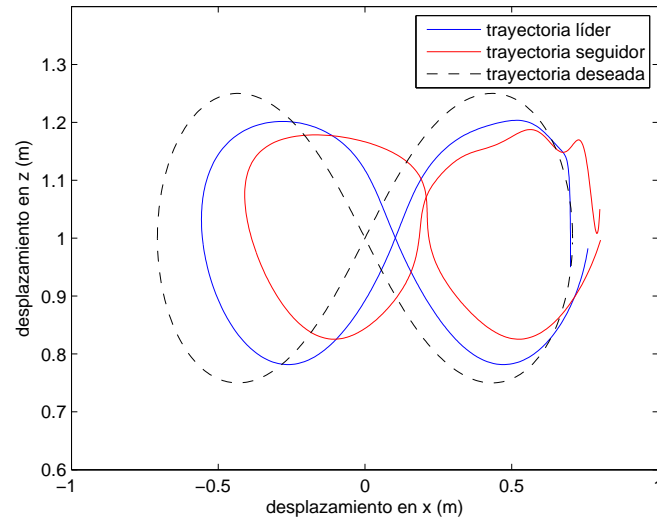
$$c_{12z} = 0$$

$$c_{12z} = 0$$

$$m_{\psi 2} = 0$$

Los resultados obtenidos de la simulación son los presentados en las Fig. 2.21, 2.22 y 2.23.

Figura 2.21: Desplazamiento en x para simulación de vuelo simultáneoFigura 2.22: Desplazamiento en z para simulación de vuelo simultáneo

Figura 2.23: Desplazamiento en el plano XZ para simulación de vuelo simultaneo

Capítulo 3

Experimentos en tiempo real

En este capítulo se presenta la descripción de la plataforma experimental usada para validar las leyes de control desarrolladas en el Capítulo anterior. Posteriormente se presenta una serie de experimentos en tiempo real referentes al vuelo individual, vuelo simultáneo con controladores desacoplados y vuelo simultáneo en configuración líder-seguidor.

3.1. Plataforma experimental.

La plataforma experimental está conformada por dos aeronaves tipo cuadricóptero AR.Drone 2.0, el sistema de posicionamiento OPTITRACK compuesto por un conjunto de 12 cámaras infrarrojas y el sistema de control y comunicación compuesto por computadoras personales. El diagrama de la plataforma experimental implementada se muestra en la Fig. 3.1.

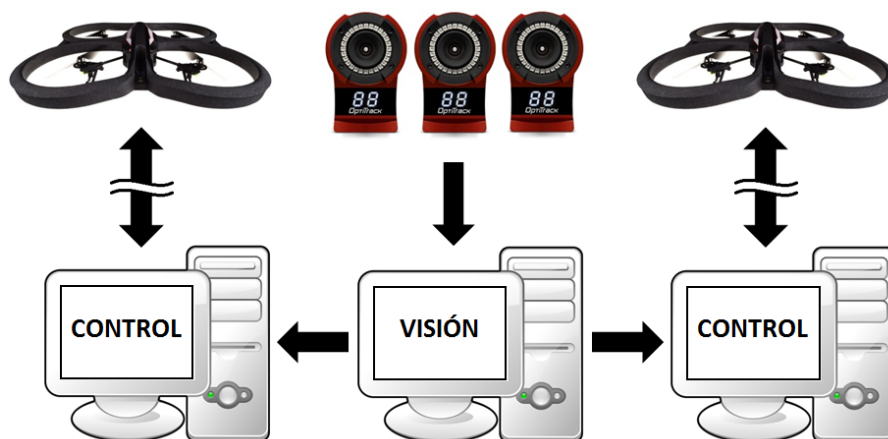


Figura 3.1: Diagrama de la plataforma experimental

La información obtenida por el conjunto de cámaras es enviada a la computadora personal encargada del procesamiento de imágenes a través del protocolo VRPN (*Virtual-Reality Peripheral Network*). Una vez realizado el procesamiento de las imágenes en la computadora encargada de la visión, los resultados obtenidos son enviados a las computadoras personales encargadas del control a través del protocolo TCP/IP (*Transmission Control Protocol/Internet Protocol*). Posteriormente las variables de control son enviadas desde las computadoras personales a los AR.Drone 2.0 de manera inalámbrica vía WIFI.

3.1.1. El AR.DRONE 2.0

El AR.DRONE 2.0 es un cuadricóptero manufacturado por la empresa francesa *Parrot*. Cuenta con un procesador ARM Cortex A8 de 32 bits a 1 GHz con DSP de video TMS320DMC64x a 800 MHz, RAM DDR2 de 1 GB a 200 MHz y opera bajo el sistema operativo Linux 2.6.32. Se comunica por medio de un módulo de Wi-Fi el cual permite controlarlo desde un teléfono inteligente, tableta o computadora personal. Tiene incorporada en la parte frontal una cámara de alta definición de 720p a 30 fps para captura de imágenes y video.

La estructura mecánica del AR.Drone 2.0 comprende cuatro rotores unidos a los cuatro extremos de una cruz a la que se unen la batería y el hardware de control. Cada par de rotores opuestos giran en el mismo sentido. Un par gira en sentido horario y el otro en sentido anti horario.

El AR.Drone 2.0 es propulsado por 4 motores de rotor interno (*inrunner*) sin escobillas de 14,5 W y 28500 rpm controlados por un microcontrolador AVR de 8 MIPS por motor. El microcontrolador detecta automáticamente el tipo de motores que se conectan y automáticamente ajusta los controles del motor. También detecta si los motores se encuentran girando o detenidos y así mismo detecta si cualquiera de las hélices se encuentra bloqueada y en caso necesario detiene todos los motores de inmediato. Este sistema de protección evita los choques repetidos.

El AR.Drone 2.0 utiliza una batería LiPo de 1000mAh a 11.1V la cual proporciona un tiempo de vuelo aproximado de 12 minutos. Mientras vuela el voltaje de la batería disminuye de carga completa a carga baja. El AR.Drone 2.0 monitorea el voltaje de la batería y convierte este voltaje en un porcentaje de la batería (12.5 volts si la batería esta completamente cargada, 9 volts si la batería presenta carga baja). Cuando el AR.Drone 2.0 detecta bajo voltaje en la batería, envía primero un mensaje de advertencia al usuario y a continuación realiza una rutina de aterrizaje automáticamente. Si el voltaje alcanza un nivel crítico el sistema completo se apaga automáticamente para evitar cualquier comportamiento inesperado.

El AR.Drone 2.0 cuenta con diversos sensores de movimiento los cuales se encuentran debajo de la carcasa central, estos son:

- Giroscopio de 3 ejes, precisión $2^\circ/s$.
- Acelerómetro de 3 ejes, precisión $+/- 50mg$ ¹
- Magnetómetro de 3 ejes, precisión 6° .

¹milésimas de la magnitud de la aceleración de la gravedad g

- Sensor de presión, precisión $+/- 10Pa$ (80cm a nivel del mar)
- Sensor ultrasónico para medición de altitud respecto al suelo
- Cámara QVGA vertical a 60 fps para medición de la velocidad respecto al suelo.

3.1.2. Sistema de posicionamiento

Para poder determinar la posición y orientación en tiempo real de cada una de las aeronaves en el espacio se utilizó el software Motive 1.7.2 junto con 12 cámaras infrarrojas Flex 13 OPTITRACK producidas por la compañía *Natural Point* (Fig. 3.2).



Figura 3.2: Cámaras infrarrojas

Cada uno de los AR.Drone 2.0 presenta en la parte superior una serie de marcadores infrarrojos, distribuidos de forma no simétrica (Fig. 3.3) con el fin de ser reconocido por el sistema de posicionamiento.



Figura 3.3: Distribución de marcadores infrarrojos sobre un AR.Drone 2.0

Las cámaras están colocadas a una altura de 4 metros cubriendo un área de 6×6 metros (Figura 3.4).



Figura 3.4: Distribución de las cámaras

La información obtenida por el sistema de cámaras se envía a través del puerto *ethernet* de la computadora de escritorio dedicada a visión (con el software MOTIVE instalado) a dos computadoras de escritorio (con el sistema operativo LINUX/UBUNTU y el AR.Drone 2.0 *Software development kit* instalados) en las cuales se implementan las leyes de control respectivas. El resultado de dichas leyes de control se envía de manera inalámbrica vía Wi-Fi a los AR.Drone 2.0.

3.1.3. Sistema de control y comunicación

El sistema de control y comunicación se encuentra conformado por 3 computadoras de escritorio.

La primera de ellas se encuentra vinculada al sistema de visión y cuenta con las siguientes características:

- Sistema operativo WINDOWS 7
- Software MOTIVE instalado

- Procesador i5 2.5 GHz
- 4 GB de memoria RAM
- 2 puertos USB 2.0 de alta velocidad para conectar las 12 cámaras
- 1 puerto USB para la llave del software Motive
- 1 puerto ethernet para comunicar con el resto de las computadoras

Las coordenadas de cada uno de los cuadrirrotores se envían vía ethernet desde la computadora encargada del sistema de visión a cada una de las dos computadoras restantes en las cuales se implementan las leyes de control de cada uno de los AR.Drone 2.0. Las características de básicas de ambas computadoras son:

- Sistema operativo LINUX, distribución UBUNTU 12.04.2 LTS (*Long Term Support*)
- API AR.Drone SDK 2.0.1 (*Software development kit*) instalado
- Procesador 2 GHz
- 2 GB de memoria RAM
- Adaptador Wi-fi para comunicar con el AR.Drone 2.0
- 1 puerto ethernet para comunicar con el resto de las computadoras

3.2. Experimentos en tiempo real

Se llevó a cabo una serie de experimentos en tiempo real con cambios en los esquemas de control y el número de agentes con el fin de comparar el comportamiento durante el vuelo. Los diferentes esquemas implementados son:

- Vuelo individual
- Vuelo simultaneo con controladores desacoplados
- Vuelo simultaneo en configuración líder-seguidor

3.2.1. Vuelo individual

Se llevan a cabo una serie de experimentos en tiempo real bajo el esquema de vuelo individual con el fin de comprobar el desempeño de las leyes de control en un ambiente de laboratorio (entorno controlado) así como para observar el comportamiento del sistema bajo efecto de perturbaciones externas. Siendo la trayectoria deseada para el plano XZ una lemniscata para $m_y = 0$ y $m_\psi = 0$ bajo condiciones iniciales arbitrarias. Las Fig. 3.5, 3.6, 3.7 y 3.8 corresponden a los resultados obtenidos. En la Figura 3.9 se presenta el seguimiento de la trayectoria deseada a lo largo del plano XZ .

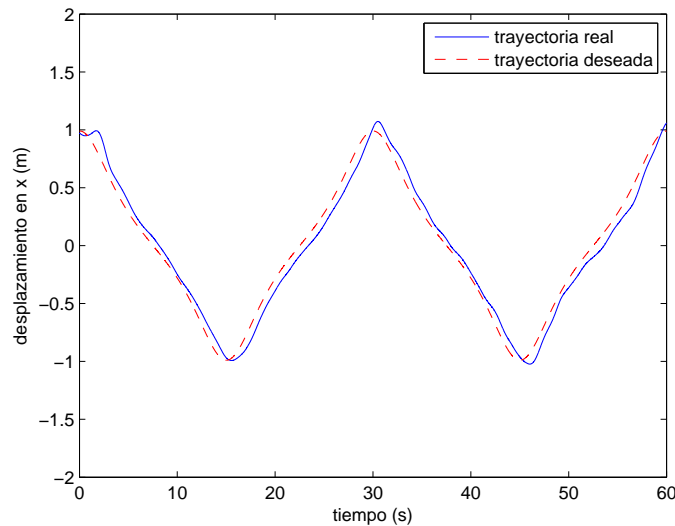
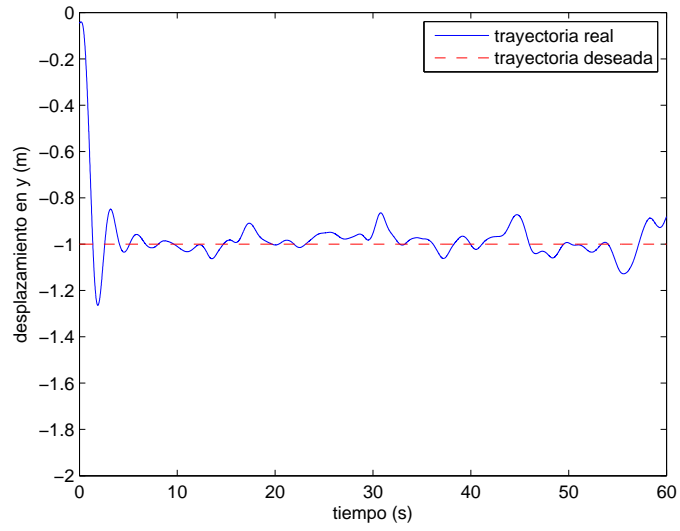
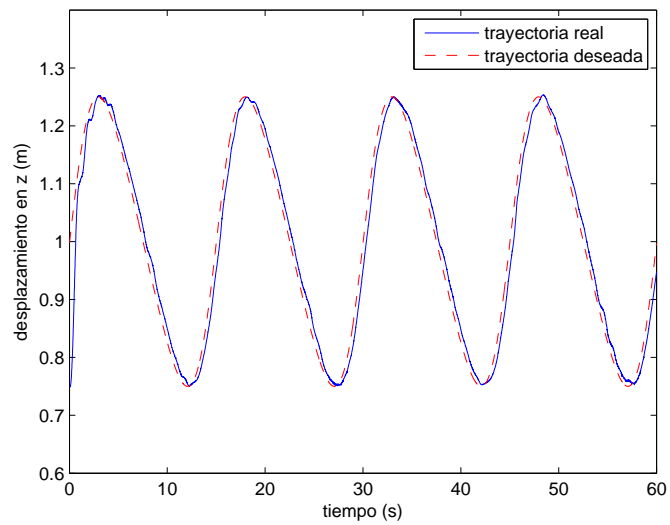
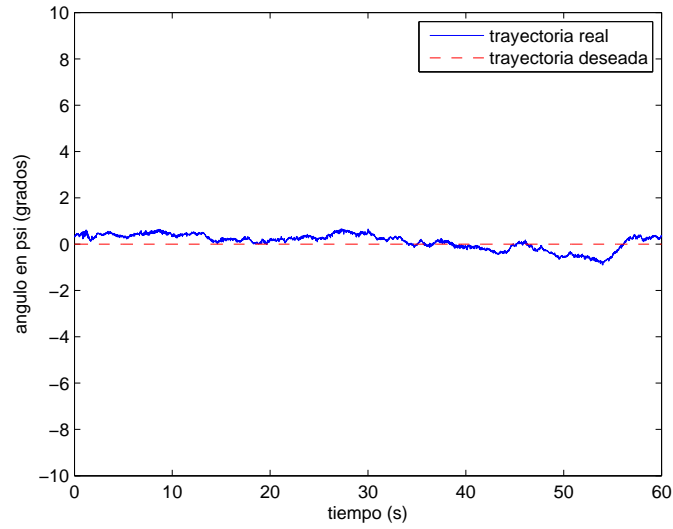
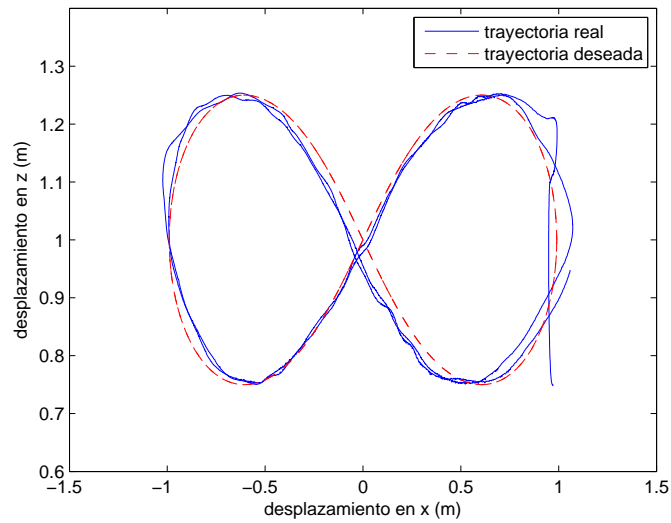


Figura 3.5: Trayectoria del cuadrirrotor en x

Figura 3.6: Trayectoria del cuadrirotor en y Figura 3.7: Trayectoria del cuadrirotor en z

Figura 3.8: Trayectoria del cuadr rotor en ψ Figura 3.9: Trayectoria del cuadr rotor en el plano XZ

Los parámetros empleados para el controlador son:

$$K_{p\theta} = 0.5$$

$$K_{d\theta} = 0.3$$

$$K_{p\phi} = 0.5$$

$$K_{d\phi} = 0.3$$

$$K_{pz} = 10$$

$$K_{dz} = 2$$

$$K_{p\psi} = 0.1$$

Los errores de seguimiento en la trayectoria se deben a la inercia del sistema, perturbaciones externas existentes como pueden ser las corrientes de aire producidas por el mismo cuadrirotor al encontrarse en un espacio cerrado.

3.2.2. Vuelo simultaneo con controladores desacoplados

Una vez validadas las leyes de control, se procede a efectuar un vuelo simultáneo de ambos agentes con controladores desacoplados siguiendo la misma trayectoria que en el caso del vuelo individual pero separados a una distancia de 2 metros entre ellos. Esto con la finalidad de observar experimentalmente los efectos creados por perturbaciones mutuas. Las gráficas de las Fig. 3.10, 3.11, 3.12 y 3.13 corresponden a los resultados obtenidos para el cuadrirotor número 1.

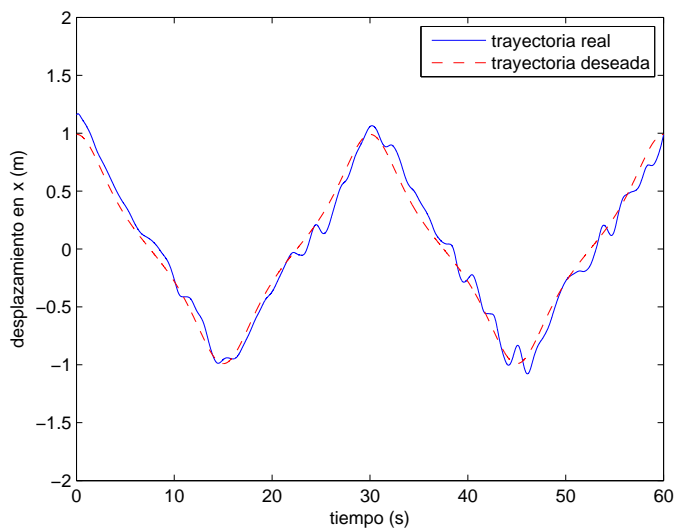
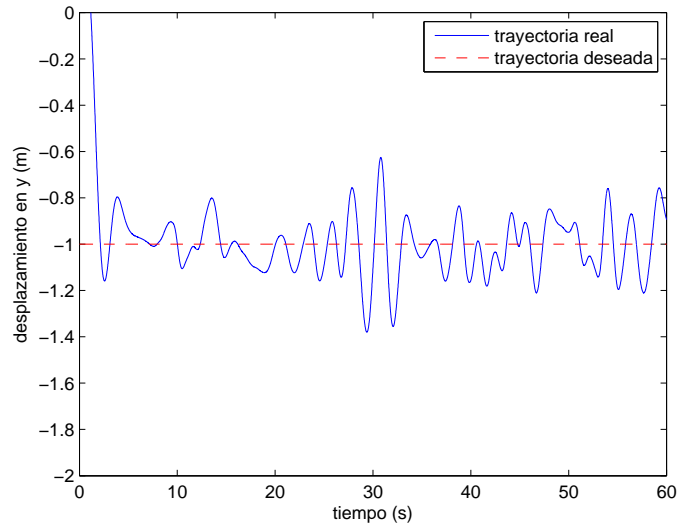
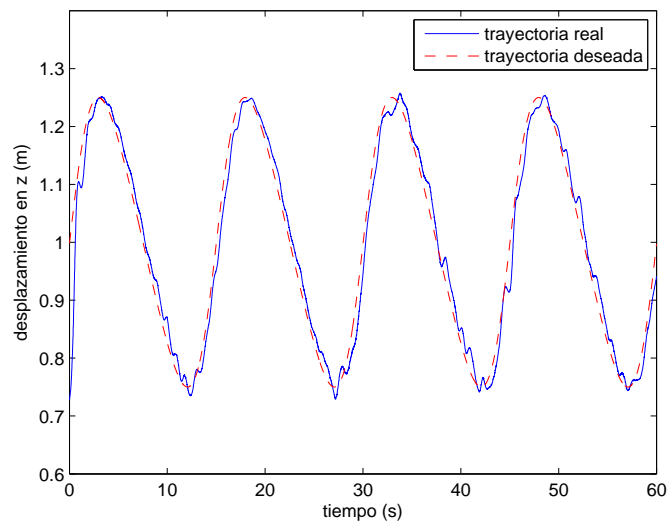
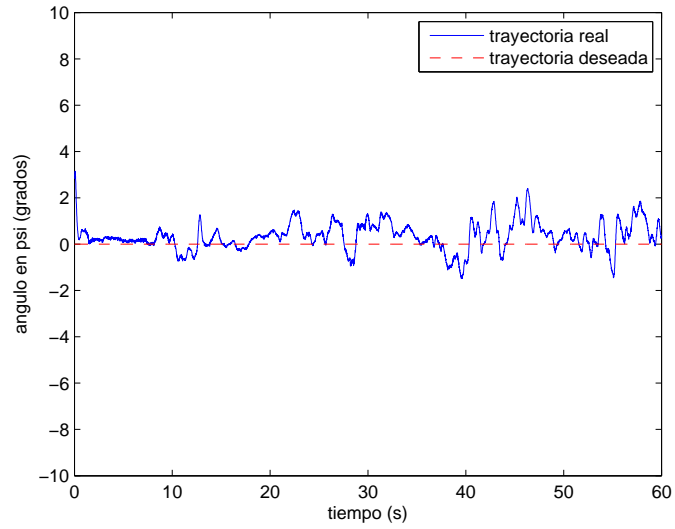
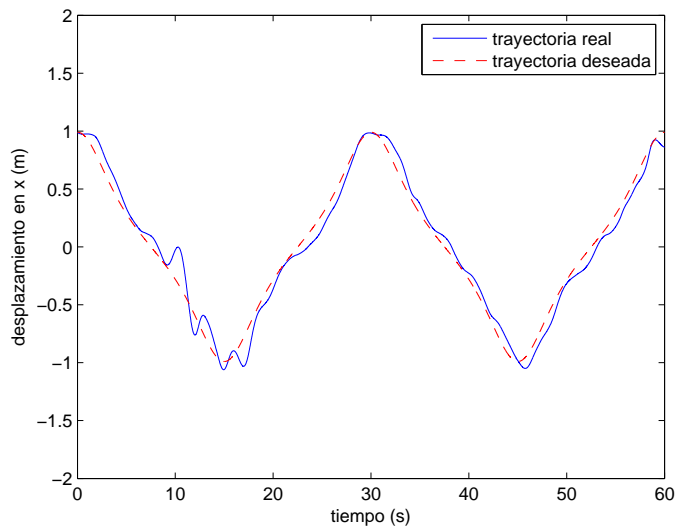


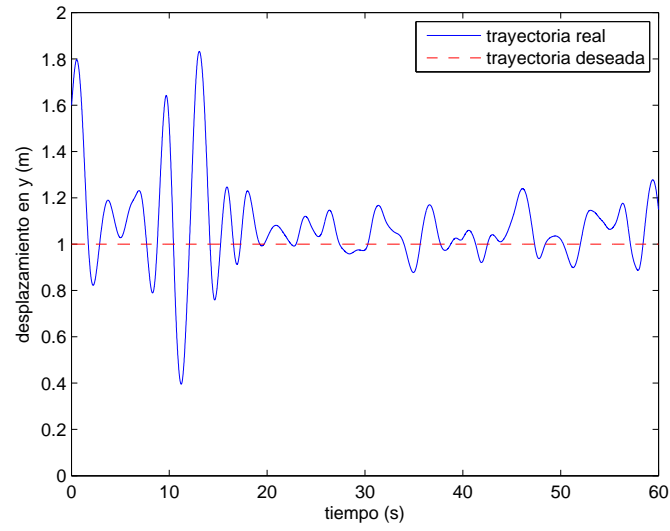
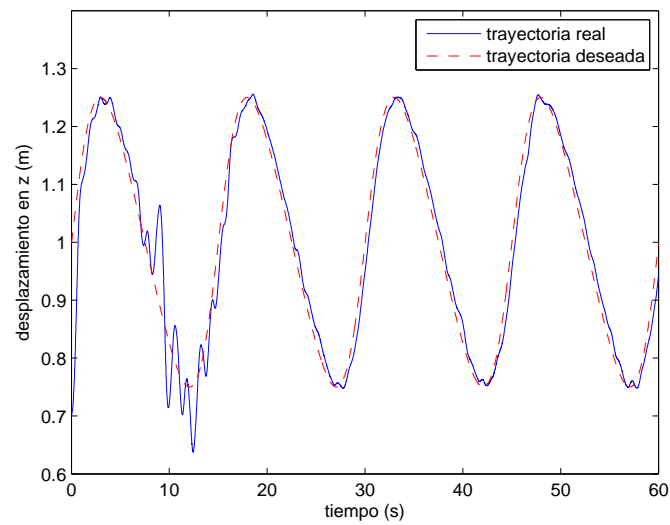
Figura 3.10: Trayectoria del cuadrirotor 1 en x

Figura 3.11: Trayectoria del cuadrirotor 1 en y Figura 3.12: Trayectoria del cuadrirotor 1 en z

Figura 3.13: Trayectoria del cuadrirotor 1 en ψ

Las gráficas de las Fig. 3.14, 3.15, 3.16 y 3.17 corresponden a los resultados obtenidos para el cuadrirotor número 2.

Figura 3.14: Trayectoria del cuadrirotor 2 en x

Figura 3.15: Trayectoria del cuadricóptero 2 en y Figura 3.16: Trayectoria del cuadricóptero 2 en z

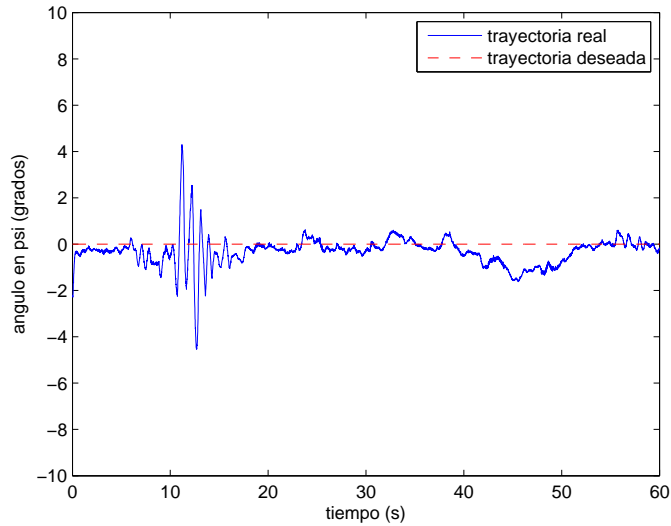


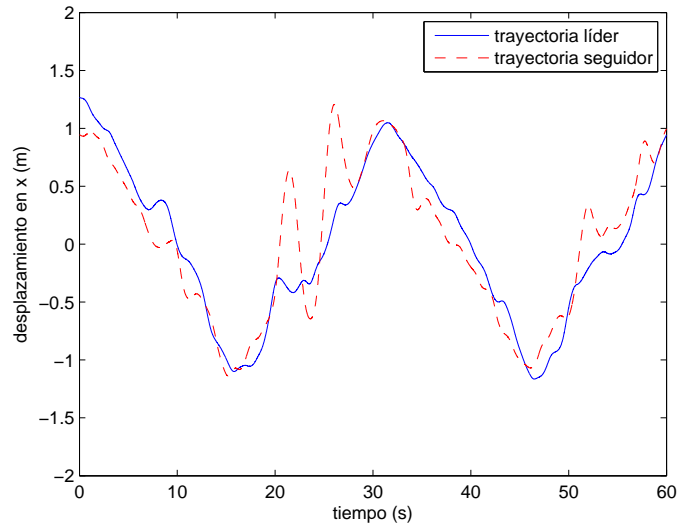
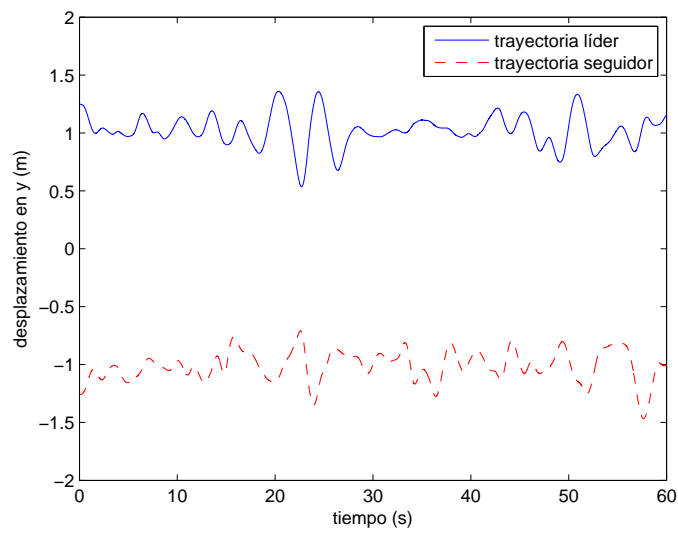
Figura 3.17: Trayectoria del cuadrirrotor 2 en ψ

Los parámetros empleados para ambos controladores son los mismos que en el vuelo individual.

Es posible observar en ambos casos los efectos generados por perturbaciones mutuas, es decir, corrientes de aire generadas por un cuadrirrotor y que afectan al otro debido a su cercanía.

3.2.3. Vuelo simultaneo en configuración líder-seguidor

Una vez observados los efectos debidos a las perturbaciones mutuas, se procede a efectuar un vuelo simultáneo de ambos agentes con los controladores presentados previamente para ejecutar el vuelo en configuración líder-seguidor. El resultado obtenido para estos experimentos se muestra en las Fig. 3.18, 3.19, 3.20 y 3.21.

Figura 3.18: Trayectorias líder-seguidor en x Figura 3.19: Trayectorias líder-seguidor en y

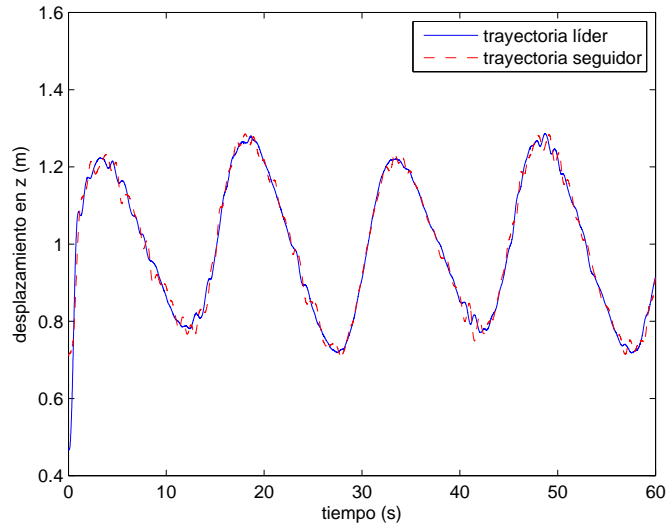


Figura 3.20: Trayectorias líder-seguidor en z

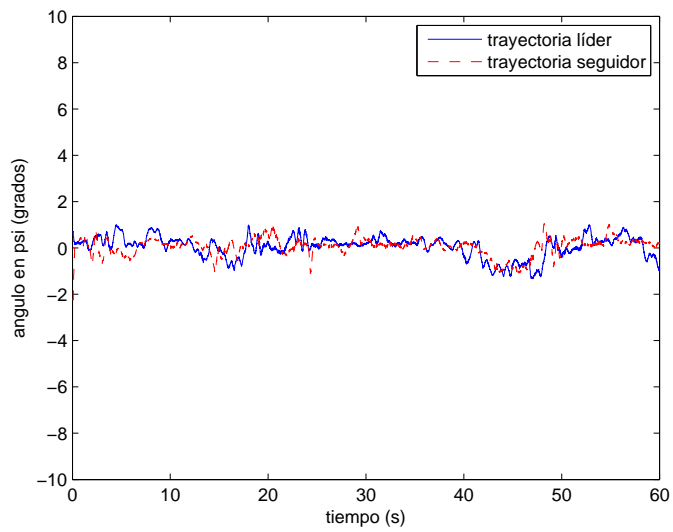


Figura 3.21: Trayectorias líder-seguidor en ψ

Los parámetros empleados para ambos controladores son los mismos que en el vuelo simultáneo con controladores desacoplados, siendo las componentes del vector de posición:

$$c_{12x} = 0$$

$$c_{12y} = 2$$

$$c_{12z} = 0$$

Se observa que a pesar de que ambos cuadrirotos se encuentran sometidos a los efectos de las perturbaciones mutuas, el agente seguidor presenta un mayor error en el seguimiento de la trayectoria. Esto es debido a que la trayectoria que el agente líder sigue se trata de una trayectoria predefinida, continua y diferenciable en todos sus puntos y el agente seguidor sigue la trayectoria resultante del agente líder, siendo esta una trayectoria no predefinida, con presencia de ruido por lo que no se garantiza que esta sea ni continua ni diferenciable en todos sus puntos.

Capítulo 4

Conclusiones y trabajo futuro

4.1. Conclusiones.

Las conclusiones a las que se llegaron en el presente trabajo son las siguientes:

- Las estrategias de control empleadas son eficientes tanto para el control de posición así como para el seguimiento de trayectorias en el AR.DRONE 2.0, pudiendo extenderse a un número mayor de agentes si el espacio de trabajo lo permite.
- El método empleado para la estimación de un modelo que describa el comportamiento del AR.Drone 2.0 es adecuado, ya que el comportamiento mostrado en las gráficas obtenidas, sin tomar en cuenta perturbaciones, es similar al comportamiento mostrado en los experimentos en tiempo real.
- El seguidor R_2 presenta un mayor error respecto a su posición debido en pequeña medida a las perturbaciones mutuas que existen entre los cuadrirotos, y en mayor medida a que la trayectoria que este debe seguir no es suave ni diferenciable en todos los puntos (a diferencia de la trayectoria deseada para el líder).

4.2. Trabajo a Futuro.

Como trabajo a futuro se plantean los siguientes puntos:

- Efectuar el seguimiento de trayectorias para una aeronave a lo largo del espacio \mathbb{R}^3 , así como la incorporación de un ángulo $\psi_d(t)$ variante en el tiempo.
- Realizar otras configuraciones referentes a sistemas multiagentes, como puede ser la formación tipo trailer o con vectores de posición variantes en el tiempo.
- Que múltiples aeronaves realicen trabajo colaborativo para una tarea que una sola no pueda efectuar, como por ejemplo el cargar con una masa de peso significativo.

Bibliografía

- [1] Real academia española. diccionario de la lengua española. 2014.
- [2] J. Ferber. Multi-agent systems. 1999.
- [3] Jorge J. Gómez Sanz. Modelado de sistemas multi-agente. volume Facultad de Informática. Universidad Complutense de Madrid, Junio 2002.
- [4] Sastry S. Hauser J. and Meyer G. Nonlinear control design for slightly non-minimum phase systems: Application to v/stol aircraft. volume Automatica Vol. 28, No. 4, pages 665–679. 1992.
- [5] V. Martínez. Modelling of the flight dynamics of a quadrotor helicopter. volume Master of Science. Cranfield, 2007.
- [6] T. Bresciani. Modelling, identification and control of a quadrotor helicopter. In *Automatic Control*, volume Master of Science, page Sweden: Lund. 2008.
- [7] E. Altug, J.P. Ostrowski, and R. Mahony. Control of a quadrotor helicopter using visual feedback. *Proc. IEEE International Conference on Robotics and Automation*, Washington, DC:72–77, 2002.
- [8] K. P. Valavanis. Advances in unmanned aerial vehicles. volume 33, page Florida. Springer, 2007.
- [9] D. Mellinger, N. Michael, and V. Kumar. Trajectory generation and control for precise aggressive maneuvers with quadrotors. In Oussama Khatib, Vijay Kumar, and Gaurav Sukhatme, editors, *Experimental Robotics*, volume 79, pages 361–373. Springer Berlin Heidelberg, 2014.

- [10] M. Achtelik, A. Bachrach, R. He, S. Prentice, and N. Roy. Stereo vision and laser odometry for autonomous helicopters in gps-denied indoor environments. *Proc. SPIE Unmanned Systems Technology XI*, 7332:Orlando, Florida, 2009.
- [11] M. Blöandsch, S. Weiss, D. Scaramuzza, and R. Siegwart. Vision based mav navigation in unknown and unstructured environments. *IEEE Int. Conf. On Robotics and Automation*, pages Anchorage, AK, 21–28, 2010.
- [12] N. Michael, J. Fink, and V. Kumar. Cooperative manipulation and transportation with aerial robots. *Autonomous Robots*, 30:73–86, 2011.
- [13] S. Al-Hiddabi. Quadrotor control using feedback linearization with dynamic extension. *Proc. 6th International Symposium on Mechatronics and its Applications, ISMA09.*, pages 1–3, 2009.
- [14] A. Chamseddine, Y. Zhang, C.A. Rabbatha, C. Join, and D. Theilliol. Flatness-based trajectory planning/replanning for a quadrotor unmanned aerial vehicle. *IEEE Transactions on Aerospace and Electronic Systems*, 48(4):2832–2848, 2012.
- [15] D. Lee, T.C. Burg, B. Xian, and D.M. Dawson. Output feedback tracking control of an underactuated quadrotor uav. *IEEE American Control Conference*, pages 1775–1780, 2007.
- [16] T. Hamel, R. Mahony, R. Lozano, and J. Ostrowski. Dynamic modelling and configuration stabilization for an x4-flyer. *15th Triennial IFAC World Congress*, 1(2), 2002.
- [17] P. Castillo, R. Lozano, and A. Dzul. Stabilization of a mini rotorcraft with four rotors. *IEEE Control Systems Magazine*, 25(6):45–55, 2005.
- [18] S.L. Waslander, G.M. Hoffmann, J.S. Jang, and C.J. Tomlin. Multi-agent quadrotor testbed control design: Integral sliding mode vs. reinforcement learning. *IEEE Intelligent Robots and Systems IEEE/RSJ*, pages 3712–3717, 2005.
- [19] G.V. Raffo, M.G. Ortega, and F.R. Rubio. An integral predictive/nonlinear structure for a quadrotor helicopter. *Automatica*, 46(1):29–39, 2010.

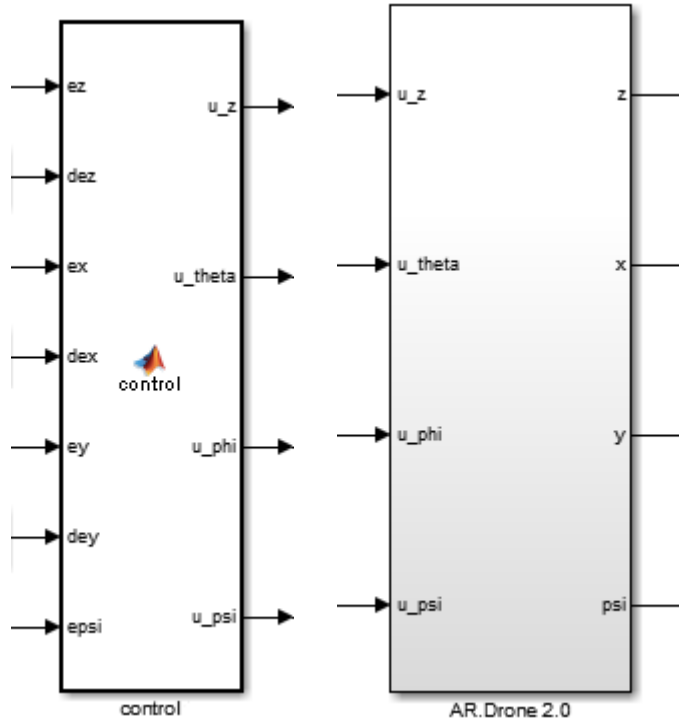
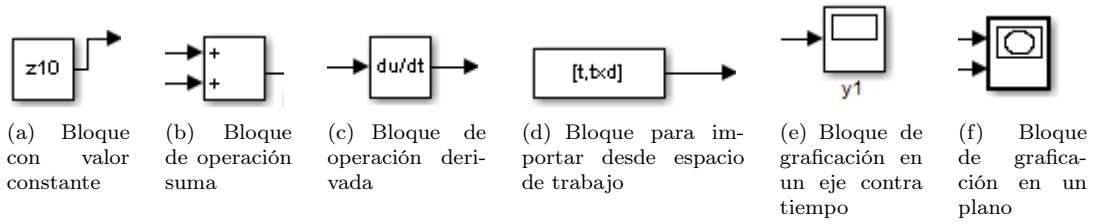
- [20] L. García-Delgado, A. Dzul, V. Santibáñez, and M. Llama. Quad-rotors formation control as an interconnected system. *Control Theory and Applications, IET*, 6(12):1787–1802, 2012.
- [21] J.J. Corona-Sánchez, J.A. Vargas-Jacob, and H. Rodríguez-Cortés. Decentralized real time implementation of a leader-follower coordination strategy for quadrotors. *Accepted to appear in the Proc. of the International Conference on Unmanned Aircraft Systems*, pages Orlando, Florida, 2014.
- [22] M. Turpin, N. Michael, and V. Kumar. Trajectory design and control for aggressive formation flight with quadrotors. *Autonomous Robots*, 33(1-2):143–156, 2012.
- [23] R. Castro, D. A. Mercado, and R. Lozano. Quadrotors flight formation control using a leader-follower approach. *In 2013 European Control Conference (ECC)*, pages 3858–3863, 2013.
- [24] David Vissière Nicolas Petit Pierre-Jean Bristeau, François Callou. The navigation and control technology inside the ar.drone micro uav. *18th IFAC World Congress*, pages Milano, Italy, 2011.
- [25] F. Belkhouche and B. Belkhouche. Modelling and controlling a robotic convoy using guidance laws strategies. *IEEE Transactions on Systems, Man and Cybernetics*, 35(4):813–825, 2005.
- [26] K. Kyriakopoulos and D. Dimarogonas. Formation control and collision avoidance for multi-agent systems and a connection between formation infeasibility and flocking behavior. *In Conference on Decision and Control*, Sevilla, Spain:84–89, 2005.

Apéndice A

Diagramas a bloques

Para llevar a cabo las simulaciones requeridas es necesario incluir los bloques necesarios así como configurarlos a modo que se describa de la forma deseada tanto al controlador como al modelo de la aeronave. Algunos de los bloques que se emplearon son:

- Bloques con valores constantes para representar condiciones iniciales (Fig. A.1(a)).
- Bloques de operación de suma (Fig. A.1(b)).
- Bloques de derivación (Fig. A.1(c)).
- Bloques para importar vectores generados desde el espacio de trabajo como son las trayectorias predefinidas (Figura A.1(d)).
- Bloques para realizar la graficación de variables a lo largo de un eje o sobre un plano (Figuras A.1(e) y A.1(f)).
- Bloques para describir funciones con múltiples entradas y múltiples salidas empleadas para implementar al controlador (Figura A.1(g)).
- Bloques de subsistemas con múltiples entradas y múltiples salidas representando al modelo de la aeronave (Figura A.1(h)).



(g) Bloque de función

(h) Bloque de subsistema

Figura A.1: Bloques empleados para la simulación

A.0.1. Bloque de control

El bloque de función es el encargado de la implementación las leyes de control (2.17) presentadas en el Capítulo 2. El contenido de dicho bloque es presentado a continuación:

```
function [u_z, u_theta, u_phi, u_psi] =
        control(ez, dez, ex, dex, ey, dey, epsi)
```

```
u_z      = -Kpz*ez      -Kdz*dez;
u_theta  = -Kptheta*ex  -Kdtheta*dex;
u_phi    = -Kpphi*ey    -Kdphi*dey;
u_psi    = -Kppsi*epsi;
```

siendo:

```
ez = z - z_d
dez = z_dot - z_dot_d
ex = x - x_d
dex = x_dot - x_dot_d
ey = y - y_d
dey = y_dot - y_dot_d
epsi = psi - psi_d
```

Se observa que la implementación de las leyes de control es directa, ya que la derivada así como el cálculo del error se llevan a cabo de forma externa. También cabe mencionar que u_z , u_θ , u_ϕ y u_ψ son solo los nombres designados a las variables resultantes de las leyes de control dejando de lado el grado de la variable, es decir, sin importar si son posiciones o velocidades.

A.0.2. Bloque AR.Drone 2.0

El bloque de subsistema es capaz de contener a otros bloques los cuales representan al modelo del AR.Drone 2.0 mediante la implementación de los siguientes bloques:

- Bloques de puertos de entrada y salida (Fig. A.2(a) y A.2(b)). Estos puertos son utilizados tanto para entrada como salida de datos al subsistema.
- Bloque de función de saturación (Fig. A.2(c)). Este bloque es empleado para limitar la salida resultante de la ley de control (entre -1 y 1), ya que de forma análoga esta acción es realizada por el controlador interno del AR.DRONE 2.0.
- Bloque de función de transferencia (Fig. A.2(d)). Este bloque es empleado para implementar en el sistema las funciones de transferencia previamente estimadas.

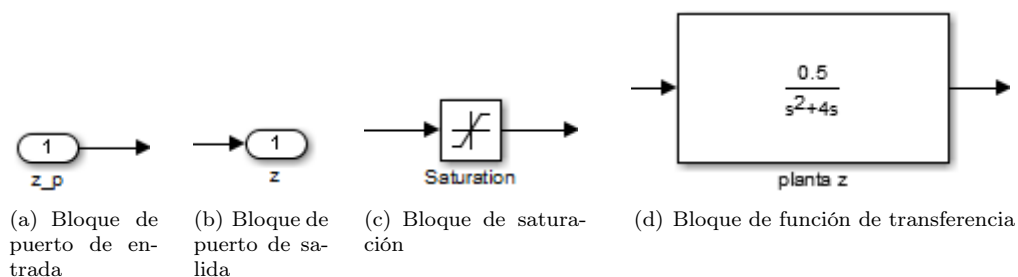


Figura A.2: Otros bloques empleados para la simulación

Al interior de este bloque encontramos con lo presentado en la Fig. A.3.

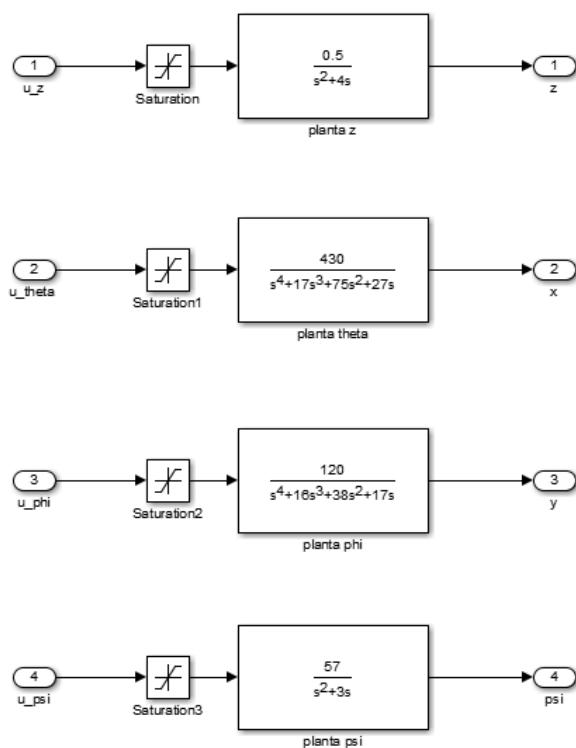


Figura A.3: Subsistema representando a un cuadrirotor

Se observan las cuatro funciones de transferencia estimadas previamente, con las que se conforma el modelo obtenido para un AR.Drone 2.0.

A.1. Diagrama a bloques para simulación de vuelo individual.

Una vez presentados todos los componentes de la simulación del sistema se presenta el diagrama a bloques completo referente al vuelo individual siguiendo una trayectoria predefinida. Dicho diagrama a bloques es presentado en la Fig. A.4.

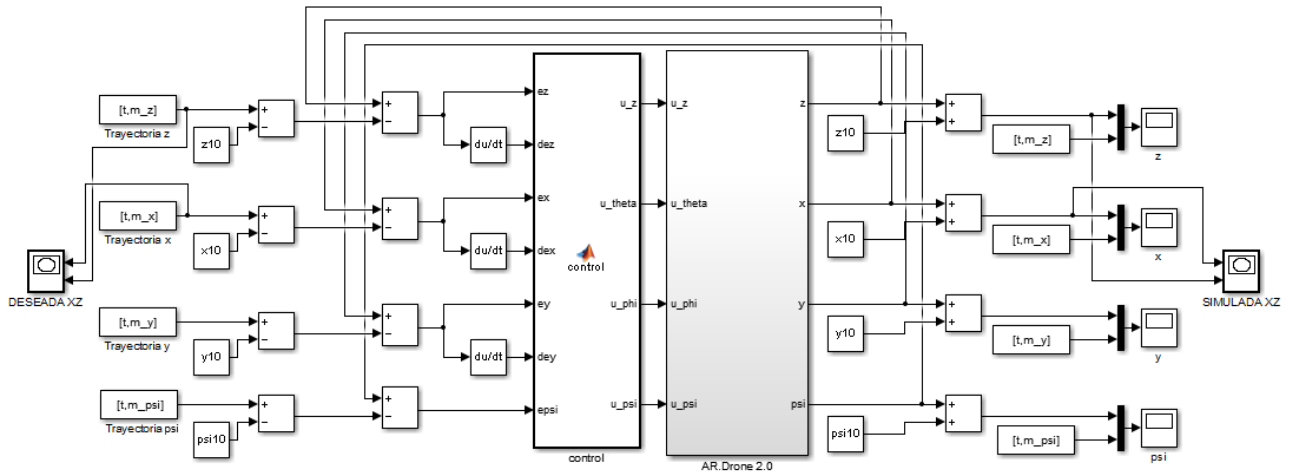


Figura A.4: Diagrama a bloques empleado para simulación de vuelo individual

A.2. Diagrama a bloques para simulación de vuelo simultaneo en configuración líder-seguidor

Los diagramas a bloques para ambos cuadrirotos permanecen prácticamente iguales, con la diferencia de que la entrada al segundo bloque de control (agente seguidor) es la salida del primer bloque (agente líder). Dicho diagrama a bloques puede ser observado en la Fig. A.5.

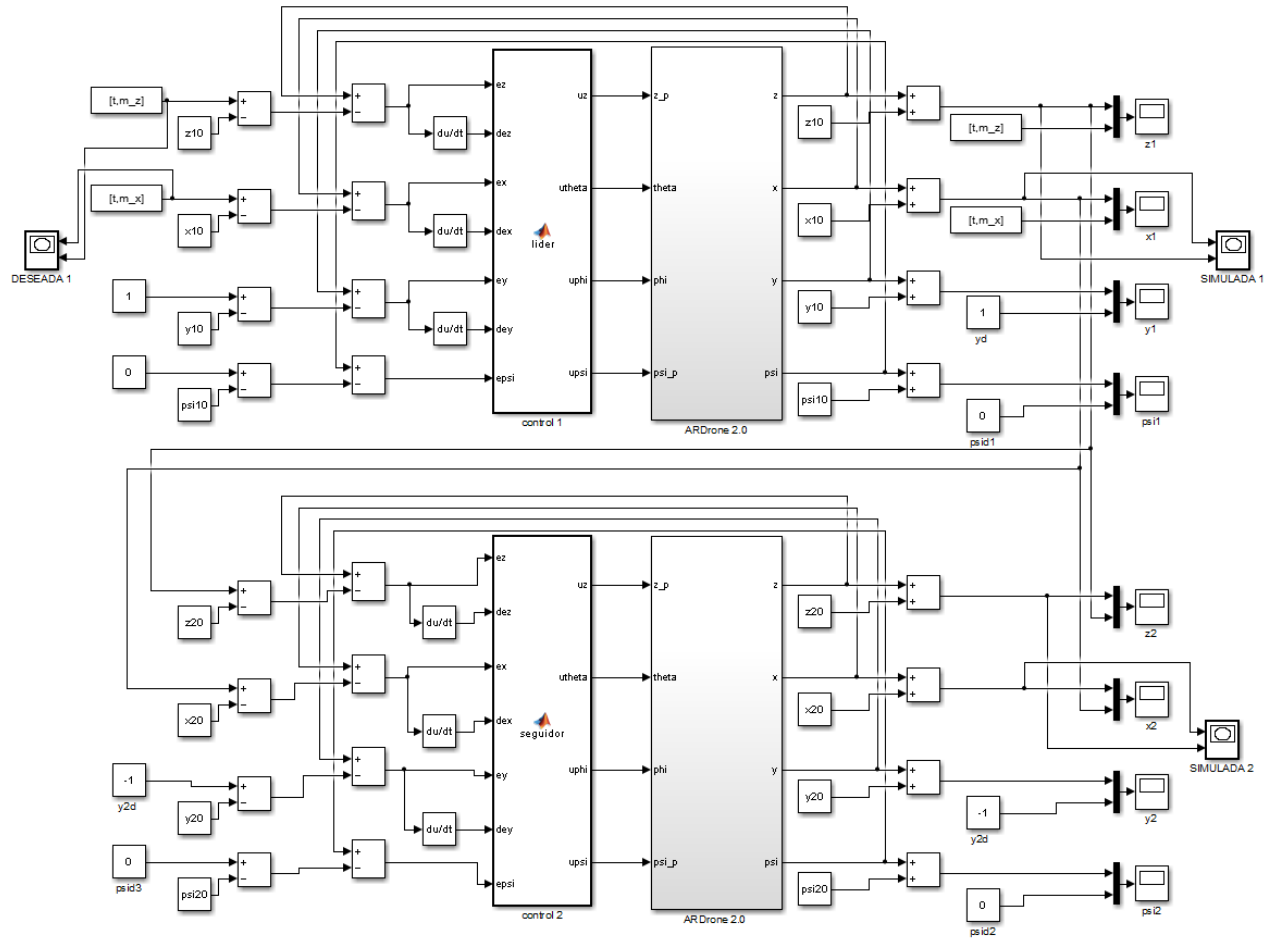


Figura A.5: Diagrama a bloques empleado para simulación de vuelo simultaneo