



# **INSTITUTO TECNOLÓGICO DE HERMOSILLO**

DIVISIÓN DE ESTUDIOS DE POSGRADO E INVESTIGACIÓN.

MAESTRÍA EN INGENIERÍA ELECTRÓNICA.

TESIS

## **IMPLEMENTACIÓN DE ALGORITMOS DE CONTROL EN UN VEHÍCULO AÉREO NO TRIPULADO.**

QUE PARA OBTENER EL TÍTULO DE:

**MAESTRÍA EN INGENIERÍA ELECTRÓNICA.**

PRESENTA:

**JESÚS RENATO MONTOYA MORALES.**

ASESOR:

**MARÍA EUSEBIA GUERRERO SÁNCHEZ.**

CO-ASESOR:

**GUILLERMO VALENCIA PALOMO.**

HERMOSILLO, SONORA, MÉXICO.

AGOSTO 2021



tecnm.mx | ith.mx





# EDUCACIÓN

SECRETARÍA DE EDUCACIÓN PÚBLICA



TECNOLÓGICO  
NACIONAL DE MÉXICO

Instituto Tecnológico de Hermosillo  
División de Estudios de Posgrado e Investigación

SECCIÓN: DIV. EST. POS. E INV.  
No. OFICIO: DEPI/158/21  
ASUNTO: AUTORIZACIÓN DE IMPRESIÓN  
DE TESIS.


06 de julio de 2021

**C. JESÚS RENATO MONTOYA MORALES,  
PRESENTE.**

Por este conducto, y en virtud de haber concluido la revisión del trabajo de tesis que lleva por nombre **"Implementación de algoritmos de control de un vehículo aéreo no tripulado"**; que presenta para el examen de grado de la MAESTRÍA EN INGENIERÍA ELECTRÓNICA, y habiéndola encontrado satisfactoria, nos permitimos comunicarle que se autoriza la impresión del mismo a efecto de que proceda el trámite de obtención de grado.

Deseándole éxito en su vida profesional, quedo de usted.

ATENTAMENTE


  
DRA. MARÍA EUSEBIA GUERRERO SÁNCHEZ  
DIRECTOR




  
GUILLERMO VALENCIA PALOMO  
CO-DIRECTOR

  
M.C. RAFAEL ARMANDO GALAZ BUSTAMANTE  
SECRETARIO

INSTITUTO TECNOLÓGICO  
DE HERMOSILLO  
DIVISIÓN DE ESTUDIOS  
DE POSGRADO

  
M.C. JOSÉ MANUEL CHÁVEZ  
VOCAL

  
M.C.C. ROSA IRENE SÁNCHEZ FERMÍN  
JEFA DE LA DIVISIÓN DE ESTUDIOS DE POSGRADO E INVESTIGACIÓN

RISF/eme\*



Av. Tecnológico S/N Col. El Sahuaro C.P. 83170 Hermosillo, Sonora  
Tel. 01 (662) 260 65 00, ext. 136, e-mail: depi\_hermosillo@tecnm.mx  
tecnm.mx | ith.mx





## CARTA CESIÓN DE DERECHOS

En la ciudad de Hermosillo Sonora a el día 20 de agosto del año 2021 el que suscribe C. JESÚS RENATO MONTOYA MORALES , alumno de la maestría en ingeniería electrónica adscrito a la División de Estudios de Posgrado e Investigación, manifiesta que es autor intelectual del presente trabajo de Tesis titulado “IMPLEMENTACIÓN DE ALGORITMOS DE CONTROL EN UN VEHÍCULO AÉREO NO TRIPULADO” bajo la dirección DE DRA. MARIA EUSEBIA GUERRERO SÁNCHEZ y ceden los derechos del mismo al Tecnológico Nacional de México/Instituto Tecnológico de Hermosillo, para su difusión, con fines académicos y de investigación.

Los usuarios de la información no deben de reproducir el contenido textual, graficas, tablas o datos contenidos sin el permiso expreso del autor y del director del trabajo. Este puede ser obtenido a la dirección de correo electrónico siguiente: m19331473@hermosillo.tecnm.mx. Una vez otorgado el permiso se deberá expresar el agradecimiento correspondiente y citar la fuente de este.

### ATENTAMENTE



JESÚS RENATO MONTOYA MORALES



# Dedicatoria

A Dios, mi familia y mis amigos.

# Agradecimientos

Agradezco a mis directores de tesis su soporte y confianza. Al comité revisor, sus oportunas retroalimentaciones. A la dirección del programa de maestría por sus gestiones y respaldó institucional. Al Instituto Tecnológico de Hermosillo, sus instalaciones, que a lo largo de estos dos años se convirtieron en un segundo hogar. A CONACYT por los recursos que se destinaron en mi preparación.

# Resumen

---

**Palabras clave:** Cuadrirrotor, controlador en tiempo real, sistema de posicionamiento, control por modos deslizantes, SLAM.

---

El presente documento de tesis tiene como objetivo el desarrollo e implementación de diferentes estrategias de control aplicadas a un vehículo aéreo no tripulado. El sistema bajo consideración es un vehículo tipo cuadrirrotor, el cual es sub-actuado debido a que tiene 6 grados de libertad, y solamente 4 grados están actuados. Se exhibe un modelo matemático bajo la formulación Newton-Euler para el sistema, se obtiene su representación linealizada considerando el primer término de la serie de Taylor y desplazamientos angulares pequeños.

Además, se muestra el diseño de un par de controladores PID y un controlador por modos deslizantes, este último se evalúa ante la presencia de perturbaciones. Se presentan los resultados en simulaciones numéricas de los controladores, para las simulaciones se toman en cuenta los parámetros físicos del vehículo AR Drone 2.0 de la marca Parrot.

También, se presenta el desarrollo y simulación de un par de controladores por modos deslizantes aplicados específicamente al vehículo AR Drone 2.0, el primero un controlador cuasi-modos deslizantes (cuasi-CMD) con una aproximación de la función signo y el segundo un cuasi-CMD asintótico. Estos controladores toman como base un modelo encontrado en la literatura de la dinámica interna del cuadrirrotor AR Drone 2.0, obtenido por la técnica identificación con el algoritmo de mínimos cuadráticos.

Se presentan un par de sistemas de monitoreo de la dinámica del móvil en tiempo real, uno basado en radio frecuencia y el segundo en una librería de SLAM (Simultaneous Localization And Mapping) visual.

Finalmente, se lleva a cabo la implementación en tiempo real sobre el cuadrirrotor previamente mencionado, se experimenta en tiempo real un controlador PD, un controlador P con control en cascada de posición y velocidad y por último un controlador por modos deslizantes. De los cuales el que presentó un mejor comportamiento fue el control P en cascada.

# Abstract

---

**Keywords:** Quadcopter, real-time controller, positioning system, sliding mode control, SLAM.

---

This thesis document aims to develop and implement different control strategies applied to an unmanned aerial vehicle. The system under consideration is a quadrotor-type vehicle, which is under-actuated because it has 6 degrees of freedom, and only 4 degrees are actuated. A mathematical model is exhibited under the Newton-Euler formulation for the system, its linearized representation is obtained considering the first term of the Taylor series and small angular displacements.

In addition, the design of a pair of PID controllers and a sliding mode controller is shown, the latter is evaluated in the presence of disturbances. The results are presented in numerical simulations of the controllers, for the simulations the physical parameters of the AR Drone 2.0 vehicle of the Parrot brand are taken into account.

Also, the development and simulation of a pair of sliding mode controllers applied specifically to the AR Drone 2.0 vehicle is presented, the first a sliding quasi-mode controller (quasi-CMD) with an approximation of the sign function and the second a quasi-mode. Asymptotic CMD. These controllers are based on a model found in the literature of the internal dynamics of the AR Drone 2.0 quadrotor, obtained by the identification technique with the least-squares algorithm.

Regarding movement systems, a couple of monitoring systems of mobile dynamics in real-time are presented, one based on radio frequency and the second in a visual SLAM (Simultaneous Localization And Mapping) library.

Finally, the implementation is carried out in real time on the previously mentioned quadrotor, a PD controller is experienced in real time, a P controller with position and speed cascade control, and finally a sliding mode controller. Of which the one with the best performance was the cascade P control.

# Índice general

<b>Dedicatoria</b>	<b>III</b>
<b>Agradecimientos</b>	<b>IV</b>
<b>Resumen</b>	<b>V</b>
<b>Abstract</b>	<b>VI</b>
<b>Índice de tablas</b>	<b>IX</b>
<b>Índice de figuras</b>	<b>X</b>
<b>Nomenclatura</b>	<b>XII</b>
<b>1. Introductorio</b>	<b>2</b>
1.1. Motivación . . . . .	2
1.2. Planteamiento del problema . . . . .	3
1.3. Hipótesis . . . . .	5
1.4. Objetivos . . . . .	5
1.4.1. Objetivo general . . . . .	5
1.4.2. Objetivos específicos . . . . .	5
1.5. Justificación . . . . .	6
1.6. Estado del arte . . . . .	6
1.6.1. Control de VANTs . . . . .	7
1.6.2. Vehículo Parrot AR Drone. . . . .	12
1.7. Metodología . . . . .	13
1.8. Organización del trabajo . . . . .	17
1.9. Conclusión . . . . .	18
<b>2. Vehículos aéreos no tripulados</b>	<b>20</b>
2.1. Introducción . . . . .	20
2.2. Clasificación de VANTs . . . . .	22
2.3. Reseña histórica. . . . .	24
2.4. Parrot AR. Drone. . . . .	25



2.5. Modelo matemático. . . . .	29
2.5.1. Descripciones geométricas. . . . .	29
2.5.2. Modelo matemático no lineal. . . . .	32
2.5.3. Modelo matemático lineal. . . . .	36
2.6. Conclusión . . . . .	39
<b>3. Estrategias de control.</b>	<b>41</b>
3.1. Control clásico. . . . .	41
3.1.1. Preliminares. . . . .	41
3.1.2. Control clásico para un cuadrirrotor . . . . .	44
3.2. Control por modos deslizantes. . . . .	52
3.2.1. Preliminares. . . . .	52
3.2.2. Control por modos deslizantes aplicado a un cuadrirrotor . . . .	54
3.2.3. Control por modos deslizantes en Parrot AR Drone. . . . .	59
3.3. Conclusión . . . . .	68
<b>4. Resultados experimentales.</b>	<b>70</b>
4.1. Plataformas de monitoreo. . . . .	70
4.1.1. Sistema por radiofrecuencia. . . . .	71
4.1.2. Visión artificial. . . . .	75
4.1.3. TUM AR Drone. . . . .	81
4.2. Implementación en tiempo real. . . . .	87
4.2.1. Control convencional. . . . .	87
4.2.2. Control CMD. . . . .	91
4.3. Conclusión . . . . .	94
<b>5. Conclusiones</b>	<b>96</b>
<b>Conclusiones</b>	<b>96</b>
5.1. Conclusiones generales . . . . .	96
5.2. Conclusiones de la experimentación . . . . .	96
5.3. Trabajos futuros . . . . .	98
<b>Referencias</b>	<b>100</b>

# Índice de tablas

1.1. Cronograma de actividades. . . . .	17
2.1. Clasificación en función de la autonomía. . . . .	23
2.2. Clasificación en función del peso promedio de despegue. . . . .	23
2.3. Comandos AT en SDK. . . . .	28
2.4. Tabla de parámetros de simulación. . . . .	37
3.1. Índices de desempeño en PID modelo linealizado. . . . .	50
3.2. Índices de desempeño en PID modelo completo. . . . .	50
3.3. Índices de desempeño en CMD. . . . .	56
3.4. Índices de desempeño en cuasi CMD. . . . .	63
3.5. Índices de desempeño en cuasi CMD asintótico. . . . .	66

# Índice de figuras

1.1.	Diagrama del pez para el problema de investigación. . . . .	14
1.2.	Metodología. . . . .	16
2.1.	Clasificación en función de las tareas despegue y aterrizaje. . . . .	22
2.2.	Movimientos del AR Drone. . . . .	25
2.3.	Carcasas protectoras Parrot AR Drone. . . . .	26
2.4.	Bosquejo de un cuadrirrotor. . . . .	29
2.5.	Ángulos de Euler. . . . .	30
2.6.	Ángulos de Euler. . . . .	31
2.7.	Movimientos en un cuadrirrotor. . . . .	32
2.8.	Sistemas de referencia. . . . .	34
2.9.	Simulación en lazo abierto para la dinámica completa y linealizada. . . . .	38
3.1.	Lazo genérico de control. . . . .	42
3.2.	Esquema de un PID. . . . .	42
3.3.	Lazo de control para dinámica linealizada. . . . .	45
3.4.	Vector de estado del PID en dinámica linealizada. . . . .	46
3.5.	Señales de control del PID en dinámica linealizada. . . . .	47
3.6.	Estructura del lazo de control. . . . .	48
3.7.	Vector de estado del PID en dinámica completa. . . . .	49
3.8.	Señales de control del PID en dinámica completa. . . . .	50
3.9.	Simulación PID con seguimiento de trayectoria. . . . .	51
3.10.	Vector de estado del CMD. . . . .	57
3.11.	Señales de control del CMD. . . . .	58
3.12.	Simulación CMD con seguimiento de trayectoria. . . . .	59
3.13.	Conceptualización del AR Drone 2.0. . . . .	60
3.14.	Posiciones del Cuasi-CMD y control P. . . . .	62
3.15.	Señales de control del Cuasi-CMD. . . . .	63
3.16.	Posiciones del Cuasi-CMD asintótico. . . . .	65
3.17.	Señales de control del Cuasi-CMD asintótico. . . . .	66
3.18.	Posiciones del Cuasi-CMD asintótico ante perturbación. . . . .	67
3.19.	Señales de control del Cuasi-CMD asintótico ante perturbación. . . . .	67

4.1.	Error en mediciones GPS Indoor. . . . .	71
4.2.	GPS Indoor HW v4.9. . . . .	72
4.3.	Monitoreo en tiempo real en Matlab. . . . .	73
4.4.	Comunicación Arduino y GPS Indoor HW v4.9. . . . .	74
4.5.	GPS Indoor en ROS. . . . .	74
4.6.	ORB-SLAM. . . . .	76
4.7.	Layout del SDK. . . . .	77
4.8.	AR Drone y sus tópicos en ROS. . . . .	80
4.9.	Implementación de ORB SLAM en AR Drone. . . . .	81
4.10.	Raspberry Pi y ORB SLAM. . . . .	82
4.11.	Ventanas de TUM AR Drone. . . . .	83
4.12.	Interfaz de TUM AR Drone. . . . .	85
4.13.	Ventanas de TUM AR Drone I. . . . .	86
4.14.	Esquema de control en tiempo real. . . . .	87
4.15.	Resultados vuelos estacionarios con control PD. . . . .	88
4.16.	Resultados vuelos estacionarios con control interno I. . . . .	89
4.17.	Resultados vuelos estacionarios con control interno II. . . . .	90
4.18.	Resultados del lazo de control con referencias distintas a cero. . . . .	91
4.19.	Control por modos deslizantes. . . . .	92
4.21.	Resultados del lazo de control en cascada. . . . .	92
4.20.	Esquema de control en cascada. . . . .	93

# Nomenclatura

## Glosario

$m$	Masa.
$I$	Tensor de inercia.
$R(\eta)$	Matriz de rotación.
$\omega$	Velocidad angular.
$r$	Vector de posición.
$v$	Velocidad lineal.
$u_i$	Empuje generado por el $i$ -ésimo rotor.
$L$	Distancia desde el centro de masa del cuadricóptero a los rotores.
$M_i$	Par generado por el $i$ -ésimo rotor en dirección de $z$ .
$u$	Empuje total.
$o$	Vector de par debido a todas las fuerzas y momentos externos.
$\eta$	Vector de orientación del móvil expresado en el marco inercial.
$\varepsilon$	Vector de velocidad angular expresado en el marco no inercial.
$k_M$	Coefficiente de arrastre.
$k_F$	Coefficiente de empuje.
$K_p$	Constante proporcional en un PID.
$K_i$	Constante integral en un PID.
$K_d$	Constante derivativa en un PID.
$t_i$	Tiempo de integración en un PID.
$t_d$	Tiempo de derivación en un PID.
$E(t)$	Señal de error.
$R(t)$	Señal de referencia.
$Y(t)$	Señal de salida.

## Lista de siglas

CMD	Controlador por Modos Deslizantes.
PD	Proporcional Derivativo.
PID	Proporcional Integral Derivativo.
VANT	Vehículo Aéreo No Tripulado.
RPY	Ángulos de balanceo (Roll), inclinación (Pitch) y orientación (Yaw).
MIE	Maestría en ingeniería Electrónica.
GPS	Sistema de posicionamiento global.
VTOL	Despegue y aterrizaje vertical.
PVTOL	Despegue y aterrizaje vertical en el plano.
LQR	Regulador cuadrático lineal.
PTAM	Seguimiento y mapeo paralelo.
IEC	Integral del Error Cuadrático.
IVAE	Integral del Valor Absoluto.
ITVAE	Integral del tiempo del Valor Absoluto .
ITEC	Integral del tiempo del Error Cuadrático.
DOF	Grado de libertad.
IMU	Unidad de masa inercial.
SDK	Kit de desarrollo de software.
LCD	Pantalla de cristal líquido.

# CAPÍTULO 1

INTRODUCTORIO

# Capítulo 1

## Introducctorio

El objetivo del presente capítulo es comunicar al lector las motivaciones e intereses que dirigieron el presente trabajo de tesis, definir el problema de investigación, así como delimitar sus alcances, metas y objetivos.

En este capítulo se presenta una breve motivación en sección 1.1. En la sección 1.2 se declara el problema de investigación, seguido de la hipótesis y objetivos presentadas en las secciones 1.3 y 1.4 respectivamente. La sección 1.5 brinda la justificación del trabajo de tesis, mientras que la sección 1.6 presenta un resumen del estado del arte mostrando de manera cronológica el tópico de investigación. La sección 1.7 introduce la metodología seguida a lo largo de la ejecución del proyecto de investigación, misma que se auxilia del diagrama de Ishikawa y de un cronograma de actividades. En la sección 1.8 se muestra una perspectiva general de la organización y contenidos de cada uno de los capítulos que constituyen el presente documento. Finalmente en la sección 1.9 se presentan unas conclusiones del capítulo.

### 1.1. Motivación

A lo largo de la última década y media se ha intensificado significativamente el estudio y aplicación de las técnicas de control en vehículos aéreos no tripulados (VANT) debido a la relevancia que han adquirido en el mercado.

Características tales su relativo bajo costo en relación a los métodos tradicionales de vuelo, a su tamaño, a la gran versatilidad en aplicaciones y la



evolución tecnológica que ha permitido el desarrollo de sistemas de control amigables para el usuario, impulsaron su éxito comercial.

Entre sus principales aplicaciones se puede mencionar: reconocimiento de patrones en campos abiertos, transporte de cargas, agricultura de precisión, monitoreo de incendios, detección de derrame de petróleo, fotografía profesional y mapeo y localización.

Para poder llevar a cabo cualquier aplicación imaginable con un cuadrirrotor, primero es necesario que se asegure su estabilidad en las tareas fundamentales de despegue, vuelo y aterrizaje, y simultáneamente que se procure la máxima autonomía en el desarrollo de ellas.

Es vital la validación experimental en el desarrollo de estas técnicas de control, debido a que brinda una perspectiva más amplia del problema, y un marco perfecto para evaluar el nivel de logro o avance.

Diversas instituciones educativas a nivel nacional imparten programas de postgrado profesionalizantes, orientados a la implementación y profundización del conocimiento en un área profesional, fomentando el uso de investigación aplicada. De manera particular, el Instituto Tecnológico de Hermosillo ofrece a la región el programa de maestría en ingeniería electrónica (MIE), misma que cuenta con dos líneas de aplicación del conocimiento: “Electrónica industrial” y “Robótica y control”. Dado el perfil profesionalizante del postgrado, brinda el marco ideal para la producción de proyectos de investigación enfocados al desarrollo tecnológico y validación experimental, esta última como ya se mencionó, vital en el estudio de técnicas de control.

## **1.2. Planteamiento del problema**

Cada día son más las aplicaciones en el mercado que implican la implementación de VANTs, su desarrollo y de las técnicas de control son las cuales son áreas del conocimiento con bastante potencial de crecimiento. Una buena parte del avance del diseño de estas estrategias llega solamente a la etapa de validación a través de la simulación, que si bien es útil es también limitada, debido a que no contempla la gran variedad de perturbaciones y condiciones a las cuales se expone el sistema en vuelo. Por lo tanto, para muchas aplicaciones es una necesidad del diseñador la validación experimental en tiempo real de la estrategia

de control, y de ser posible, esta validación debe contemplar la dinámica completa del sistema.

Por otra parte, las plataformas comerciales de desarrollo de VANTs cuentan con una retroalimentación interna, pero no siempre se encuentran habilitadas para lectura y escritura externa, y estos sistemas de retroalimentación, por lo general sistemas de posicionamiento global (GPS), están diseñados para trabajar en exteriores por lo que pierden su precisión en espacios cerrados. Los laboratorios oficiales de desarrollo que cuentan con sistemas de retroalimentación para los 6 grados de libertad a través de visión, por ejemplo, el sistema OptiTrack o Vicon, los cuales poseen la precisión y tasa de transferencia de información adecuada para esta tarea, son muy escasos en nuestro país debido al alto costo de estos equipos.

De manera particular, actualmente en el Instituto Tecnológico de Hermosillo no se cuenta con una plataforma validada sobre la cual experimentar las estrategias de control, y en investigaciones locales anteriores se logró el diseño de un sistema de monitoreo para la dinámica rotacional del cuadrirrotor, pero deja descubierta la dinámica traslacional y el control en lazo cerrado.

A la hora de la implementación en tiempo real, el investigador se encuentra ante un conjunto de retos tales como la lectura y manejo de la información proveniente de los sensores, que en la mayoría de los casos requiere del uso de filtros; la programación del sistema de control en un sistema embebido; el control de los actuadores que cada vez son más sofisticados; la comunicación en tiempo real con la estación de control a una tasa adecuada, y el efecto de las dinámicas no modeladas y perturbaciones, solo por mencionar algunos.

Ya se comentó a grandes rasgos la importancia de implementar estrategias de control en VANTs, debido a la relevancia científica y al mercado potencial por explorar, también se comentaron posibles retos con los cuales se enfrentará el investigador, por lo tanto, surge la pregunta de investigación, misma que define el problema y la razón de la presente trabajo de tesis.

¿Es posible implementar estrategias de control modernas para la estabilización de la dinámica completa de un vehículo aéreo no tripulado, a través del uso de técnicas de programación y de plataformas de desarrollo de bajo costo, ajustándose al contexto institucional?

### 1.3. Hipótesis

Por medio del uso, manejo y manipulación de una plataforma del VANT y de entornos de programación y desarrollo para sistemas robóticos, el sistema de posicionamiento por radio frecuencia y/o visión artificial se podrán implementar estrategias de control para estabilizar la dinámica completa del vehículo.

### 1.4. Objetivos

En este tema de tesis se tienen los siguientes objetivos:

#### 1.4.1 Objetivo general

Implementar algoritmos de control, con el propósito de estabilizar la dinámica completa del cuadricóptero AR Drone 2.0, en tiempo real, a través de entornos de programación y desarrollo para sistemas robóticos auxiliado de tarjetas de desarrollo de bajo costo.

#### 1.4.2 Objetivos específicos

- Obtener una representación no lineal para la dinámica completa del cuadricóptero y simular en lazo abierto.
- Obtener y simular una representación lineal equivalente para la dinámica del cuadricóptero.
- Diseñar y simular un controlador proporcional-derivativo (PD) para estabilizar el vuelo estacionario del cuadricóptero.
- Desarrollar un sistema de monitoreo y retroalimentación para los 6 grados de libertad del VANT auxiliado del sistema de navegación en interiores Marvelmind set HW4.9 (basado en radiofrecuencia), y/o visión artificial.
- Caracterizar e implementar en tiempo real el controlador proporcional derivativo en el AR Drone 2.0 marca Parrot.

- Diseñar y simular un controlador no lineal mediante la técnica de modos deslizantes con el propósito de estabilizar la dinámica completa del cuadrirrotor.
- Caracterizar e implementar en tiempo real el controlador no lineal diseñado en el AR Drone 2.0 estabilizando solamente la dinámica rotacional.
- Caracterizar e implementar en tiempo real un controlador no lineal diseñado en el AR Drone 2.0 estabilizando su dinámica completa.

## 1.5. Justificación

El presente trabajo de investigación tiene como finalidad la validación experimental de técnicas de control en cuadrirrotores contemplando los retos que esta presenta, sometidos al contexto institucional, y de esta manera contribuir al desarrollo tecnológico de la región.

Su desarrollo y culminación contribuiría al ecosistema de investigación del estado de Sonora, con la publicación de un artículo en la revista de divulgación estatal AVIIES y con la impartición de conferencias a nivel licenciatura de los avances del trabajo de tesis.

Además, genera contribuciones a la institución, debido a que en la etapa final del trabajo de tesis se validó una plataforma de experimentación con su respectivo sistema de retroalimentación, lo cual habilita una vía para el desarrollo de tareas más complejas empleando control de cuadrirrotores.

## 1.6. Estado del arte

El empleo de vehículos aéreos no tripulados, como se conocen en la actualidad, data a partir de la primera guerra mundial donde tuvieron aplicaciones militares como objetivos de entrenamiento, en tareas de reconocimiento, en lanzamiento de proyectiles, incluso en la distribución de propaganda.

A lo largo de la última década y media se ha visto en auge la aplicación del VANT en tareas civiles, en ámbitos como la ingeniería civil, pesca, agricultura, fotografía y en entretenimiento, con esto ha aumentado la investigación sobre técnicas de

control que permitan una mayor autonomía y mejor respuesta a las condiciones dinámicas a la que se ve expuesto.

### 1.6.1 Control de VANTS

En [1], se implementa un controlador predictivo basado en retroalimentación de estados en tiempo discreto, demostrando una estabilidad robusta y convergencia pese a la incertidumbre en los parámetros de la planta, el retardo del sistema y el periodo de muestreo. En [2], se usa un modelo basado en lagrangianos y una técnica de control basado en el análisis de Lyapunov con componentes proporcionales y derivados del error, se estabiliza la dinámica de un cuadrirrotor en las tareas de vuelo estacionario y aterrizaje, donde gracias al uso de la teoría de control no lineal se posibilitó el seguimiento de trayectorias más agresivas.

En [3], se introduce la experimentación auxiliada de técnicas de visión artificial, se estabiliza el despegue de un vehículo aéreo de despegue y aterrizaje vertical "PVTOL" usando una cámara para medir la posición y la orientación del PVTOL que se mueve en un plano inclinado. Se utilizó una estrategia de control lineal simple para estabilizar el sistema con el fin de facilitar los experimentos reales. La ley de control propuesta asegura la convergencia del estado con el origen. En [4], se presenta un algoritmo de control simple para estabilizar la rotación del cuadrirrotor, utilizando la técnica de retroalimentación y agregando funciones de saturación. El análisis de convergencia se llevó a cabo utilizando el análisis de Lyapunov. Se comparó el rendimiento del controlador con respecto a un controlador PD clásico mediante simulación, además la robustez del algoritmo de control con respecto a las perturbaciones agresivas se ilustró con experimentos en tiempo real.

[6] muestra el diseño y control de un cuadrirrotor, se diseña un modelo basado en las ecuaciones de Newton-Euler y utiliza un modelo de la literatura linealizado en el punto de operación para los motores y hélices. En este trabajo los autores proponen un control con base a un análisis de Lyapunov implementándolo en un microcontrolador PIC, sin embargo, solamente corroboran el control de rotación. En [7], se realiza la comparación de un PID y un control óptimo adaptativo, uno para la dinámica linealizada y otro para el modelo completo basados en un análisis de energía, ambos se implementan en tiempo real solamente controlando la orientación. Posteriormente, [8] desarrolla la comparación de un control por modos

deslizantes y un backstepping, donde con un modelo basado en energía y un análisis de Lyapunov se diseñan ambos controladores, también, este trabajo presenta simulaciones para la dinámica completa e implementan en tiempo real solamente el control de orientación.

En [9], se simula un control integral backstepping para la dinámica completa, considerando el modelo de los actuadores, se simula el seguimiento de trayectorias y la evasión de obstáculos e implementan vuelos estacionarios. Los trabajos anteriores son importantes debido a que abrieron una beta de investigación en este tópico, y sirvieron como base para las investigaciones posteriores.

En [10], se presenta el diseño de un sistema de control integrado de piloto automático para aeronaves de despegue y aterrizaje vertical (VTOL) utilizando sensores de bajo costo. El sistema de control integrado utiliza una arquitectura de procesamiento paralelo. Además, se utiliza software multitarea para implementar la adquisición de datos, el cálculo de la ley de control y la salida de corrección para obtener el punto de ajuste deseado. La ley de control se puede ajustar fácilmente para mejorar el rendimiento del vehículo. Se evaluó el rendimiento de esta plataforma en un helicóptero de 4 rotores. El objetivo principal es lograr el vuelo estacionario utilizando dos estrategias de control, un control de PD lineal y un control de saturaciones anidadas no lineales. Los experimentos en tiempo real muestran que el piloto automático es una plataforma confiable con componentes de bajo costo. [11] presenta resultados para calcular el margen de robustez de un sistema de control de rotación de un mini-helicóptero de 4 rotores. La incertidumbre paramétrica máxima se calcula cuando se utiliza un control PD multivariable para estabilizar la actitud del vehículo aéreo. Este trabajo se basa en la caracterización del conjunto de valores del modelo matemático para el sistema de control. Este modelo matemático está representado por una planta de intervalo con retardo de tiempo. El principio de exclusión cero se utiliza para calcular el margen de robustez del sistema de control de lazo cerrado. Este enfoque transforma el problema original de estabilidad robusta en un simple problema de inspección gráfica donde solamente necesitamos verificar si una gráfica en el plano complejo contiene el origen o no. Además, se presentan experimentos en tiempo real que muestran el desempeño satisfactorio de la estrategia de control propuesta.

La navegación autónoma de un vehículo aéreo no tripulado se puede lograr con un sistema reactivo que permite al robot superar todos los cambios inesperados en

su entorno. En [12], se propone un enfoque para evitar obstáculos frontales utilizando propiedades conocidas del flujo óptico y aprovechando la capacidad de vuelo estacionario del helicóptero. Se propone una máquina de estados como solución para dotar al VANT de todas las reacciones necesarias para la navegación interior. Se muestra cómo se pueden lograr transiciones suaves disminuyendo la velocidad del vehículo proporcionalmente a la distancia de un obstáculo y por breves instantes de vuelo estacionario. En [13], se propone un método de control de posición basado en visión, con el propósito de proporcionar cierto nivel de autonomía a un vehículo aéreo no tripulado de 4 rotores. Este enfoque estima la posición  $x$ ,  $y$  y  $z$  del helicóptero con respecto a una plataforma de aterrizaje en el suelo. Esta técnica permite medir las variables de posición que son difíciles de calcular cuando se utilizan sistemas de navegación convencionales, por ejemplo, sensores inerciales o sistemas GPS. También se presenta un método para medir la velocidad de traslación en un marco local. La estrategia de control implementada se basa en un controlador de retroalimentación de estado completo. Los resultados experimentales validan la eficacia del método.

En [14] se muestra un modelo dinámico no lineal para la rotación basado en cuaterniones, así como su correspondiente controlador de ganancia LQR. Todas las consideraciones para los movimientos del cuadirrotor se describen a través de sus variables de estado. [15] presenta el concepto de parametrización de tiempo para maniobras de vuelo del cuadirrotor que establece un controlador de retroalimentación robusto y sin modelo basado en una representación de cuaternión sin singularidades. Se destacan tres características: el algoritmo de control asegura la estabilidad exponencial de la dinámica de posición / orientación completa del sistema con esfuerzos de control suaves. El sistema de lazo cerrado es robusto en presencia de fuerzas externas y momentos inducidos generados durante las maniobras de vuelo. El cuadirrotor controlado ofrece capacidades para maniobras agresivas. El modo deslizante da lugar a una estabilidad terminal bien planteada, parametrizada por un tiempo de convergencia deseado definido por el usuario.

En [16] se sintetiza una estrategia de control de estabilización sub-óptima no lineal basada en las funciones de control Lyapunov (CLF) y se aplica en un cuadirrotor. Se obtuvieron condiciones para garantizar la estabilidad asintótica del sistema en lazo cerrado. Usando la CLF candidata, se obtiene una ley de control

óptima de estabilización no lineal que permite el ahorro de energía. Se desarrollan simulaciones numéricas tanto para estrategias de control como para experimentos en tiempo real utilizando el algoritmo de control de estabilización no lineal. [17] se enfoca en el problema del casteo del control de modo deslizantes. Se propone un esquema de control de altitud robusto en tiempo real para el rendimiento eficiente del cuadricóptero. La detección de la medición de altitud se realiza mediante un sensor de presión para obtener un control robusto de la altitud del vehículo en modo de desplazamiento, tanto en interiores como en exteriores. La medición de altitud tiene la ventaja de introducir esta información de estado directamente en el control de lazo cerrado que debería ser muy útil para lograr una estabilización robusta del control de altitud, lo que eliminó el parateo. El control de modo deslizante propuesto elimina el fenómeno de parateo al reemplazar una función de signo con una función de saturación de alta pendiente.

En [18] se realiza un control por modo deslizante sin modelo basado en una superficie de deslizamiento de orden fraccional, se basa en: i) una estructura de control absolutamente continua que no requiere el modelo dinámico exacto para inducir un movimiento de deslizamiento fraccional en tiempo finito, ii) una metodología para diseño fraccional. Se proponen referencias con una contraparte clara en el dominio de la frecuencia, esto para mejorar la respuesta del sistema, en particular el período transitorio, y generar un alto rendimiento durante el movimiento deslizante. [19] propone un método robusto de control no lineal para la orientación de cuadricópteros robóticos inciertos. El controlador propuesto se desarrolla en base a un modelo no lineal con la representación del cuaternión y sujeto a incertidumbres de parámetros, no linealidades y perturbaciones externas. Se propone un controlador de retroalimentación de estado para restringir los efectos de las no linealidades e incertidumbres en el sistema de control de lazo cerrado. Se compone de un controlador nominal y un compensador robusto. El controlador nominal determina los polos deseados del sistema de control de lazo cerrado nominal y se introduce el compensador robusto para atenuar los efectos de las perturbaciones equivalentes de entrada.

En [20] se diseña y valida un esquema de control de cuaterniones para estabilizar globalmente un cuadricóptero. Se propone una ley de control de actitud para estabilizar la trayectoria del vehículo, luego se desarrolló una ley de control de posición para estabilizar el vehículo alrededor del punto deseado. Usando



referencias de posición, se calculó una trayectoria suave para el controlador de actitud de modo que la posición del vehículo se estabilice. El algoritmo de control propuesto permitió el comportamiento de un sistema cuadirrotor lineal. El modelo dinámico se simula en lenguaje de programación Python y se estabiliza con éxito utilizando un controlador PD. Solamente se implementa la estabilización de la velocidad de traslación en la plataforma experimental. [21] propone el seguimiento simultáneo de la posición y trayectoria variable en el tiempo, basado en un control de modo deslizante integral con una superficie deslizante basada en el cuaternión. Se obtiene un seguimiento exponencial sin requerir ningún conocimiento del modelo dinámico o sus parámetros para la implementación. Dado que se aplica un múltiple de error de orientación invariante lineal todo el tiempo, se introduce una ganancia variable en el tiempo para una convergencia de tiempo finita bien planteada. Se exploran aplicaciones ilustrativas en un estudio de simulación que muestra la viabilidad y versatilidad del seguimiento de posición-guiñada en la vigilancia de un objetivo de campo de visión (FoV), destornillador aéreo y agarre aéreo.

En [23] se presenta una estrategia de control que se basa en un método de servo visual mejorado basado en imágenes con una ganancia de servo dinámico ajustada por algoritmo genético (IBVS-GA). Cumpliendo con las características sub-actuadas del cuadirrotor, el modelo simplificado IBVS se diseña donde la dimensión de los parámetros de control se reduce a 4 de 6. El algoritmo genético se utiliza para obtener el valor de ganancia de servo apropiada. En [24], se emplea un controlador difuso para sintonizar los tres parámetros del controlador PID en la altitud y en la orientación. Además, presenta en detalle la realización del algoritmo de control propuesto para el cuadirrotor. La prueba de simulación sobre el seguimiento de altitud y rotación del VANT se completa en Matlab / Simulink usando el controlador PID y el controlador PID difuso, respectivamente. Los resultados experimentales muestran que el PID junto con el control difuso tiene un mejor rendimiento de seguimiento que el PID puro.

En [25], se combina una red inalámbrica de última generación y un vehículo aéreo no tripulado para crear un sistema de control de vuelo cuadirrotor en tiempo real. Se propone un algoritmo de control tolerante a fallas adaptativo robusto, distribuido y jerárquico, basado en una teoría robusta y tolerante a fallas para mejorar el rendimiento de este sistema. La simulación y los resultados de pruebas

experimentales muestran que cuando este sistema se analiza con respecto a perturbaciones externas, fallas internas del actuador y fallas de asociación de la red inalámbrica, la estabilidad del controlador es asintóticamente estable.

### 1.6.2 Vehículo Parrot AR Drone.

De manera particular, en el cuadricóptero AR Drone como plataforma de experimentación se han implementado distintas estrategias de control aprovechando el control interno que este presenta, entre las cuales se encuentran el controlador PID, PID difuso, backstepping y control basado en pasividad.

Por ejemplo, en [26], se implementa un control PID para regulación a partir de la identificación de sus parámetros, obteniendo un modelo matemático aproximado a su dinámica interna en el dominio de la frecuencia. Un trabajo similar se muestra en [31], donde se auxilian de la paquetería de Matlab para identificar los parámetros de la planta, pero solamente se simula el controlador PID. En [32] se utiliza una señal binaria pseudo-aleatoria para identificar los parámetros del cuadricóptero, sin embargo, los resultados del controlador presentan errores muy grandes. En [29] se presenta otra identificación de la dinámica interna, y con base a ella en [30] se implementa una combinación de un control adaptativo neuro difuso con un PID.

En [27] y [28] se implementa mediante LabView dos esquemas distintos para un control lógico difuso, el primero es un esquema de control que contempla la posición en  $x$  y  $y$  mediante el ángulo de orientación  $\psi$  y el ángulo de inclinación  $\phi$  y solo realiza una compensación en el ángulo de balanceo  $\theta$ . En el segundo esquema de control,  $x$  y  $y$  dependen del ángulo de inclinación  $\phi$  y del ángulo de balanceo  $\theta$ , mientras que  $\psi$  se mantiene constante, siendo este último esquema significativamente más rápido que el primero. En [37] se presenta un controlador PID difuso, donde para la retroalimentación de estados se utiliza el sistema de monitoreo y mapeo simultáneo (PTAM), se realizan experimentos de vuelo estacionario a distintas alturas, obteniendo resultados similares a los que se obtienen con el control interno en esta misma tarea.

En [33] se realiza progresivamente una identificación de parámetros mediante la técnica de mínimos cuadrados, auxiliados de un OptiTrack y un controlador por asignación de polos con seguimiento de trayectorias. En [34], con base a la identificación previa, se propone e implementa un controlador backstepping con

seguimiento de trayectorias. Por último, los autores presentan en [35] una extensión de [34] realizando una implementación multi-agentes de la técnica de control.

En [36] se presenta un control predictivo con prevención de colisiones, donde para la identificación se auxilian del sistema OptiTrack y se modela a partir de la respuesta en lazo abierto a la función escalón. Se construye un modelo semi-lineal sobre el cual se propone un control predictivo basado en modelo (MPC). El sistema de control es capaz de detectar obstáculos y ajustar su trayectoria respecto a estos, se presentan resultados experimentales del sistema de control. En [37] se propone un control basado en pasividad y un control de reinicio para el AR Drone. El modelo se identifica a partir de datos experimentales de entrada y salida presentados en la literatura, el modelo generado de planta es pasificado por compensación de ganancia no lineal y cancelación de polo-cero. Finalmente, el modelo de planta pasificado se estabiliza por un controlador de reinicio.

## 1.7. Metodología

Para poder dar respuesta al problema de investigación del presente trabajo se realiza una lluvia de ideas y se construye el diagrama de Ishikawa mostrado en la Figura 1.1, y posteriormente se declaran los objetivos mostrados en la Sección 1.3. Habiéndose declarado los objetivos se construyó el diagrama mostrado en la Figura 1.2, en el que se muestra la correlación y el flujo de las tareas que dan respuesta a los objetivos específicos, y mediante el cual se determinó su prioridad y jerarquía. Se propuso un cronograma mostrado en la Tabla 1.1, con el cual se muestran las principales actividades realizadas cada semestre.

La primera actividad fue una inmersión inicial al tópico de investigación, se realizó una primera búsqueda del estado del arte, se identificaron las posibles plataformas a validar, trabajos previos sobre el tema de estudio y paqueterías de software. Hecho lo anterior se realiza la segunda actividad: la propuesta de tesis, en la cual se definió el título, los objetivos, las metas y alcances del tema de tesis.

Las tareas de modelado matemático, configuración de la plataforma de VANT, y diseño del sistema de monitoreo se ejecutan en paralelo a partir del segundo semestre. Los productos para dicho periodo fueron el modelado matemático y su representación lineal con sus respectivas simulaciones numéricas. Se toma como referencia un trabajo del modelo matemático presente en la literatura en la

formulación Newton-Euler, se realiza una segunda búsqueda del estado del arte, para identificar las características físicas del VANT seleccionado y con base a éstas se simula el comportamiento de la dinámica en lazo abierto.

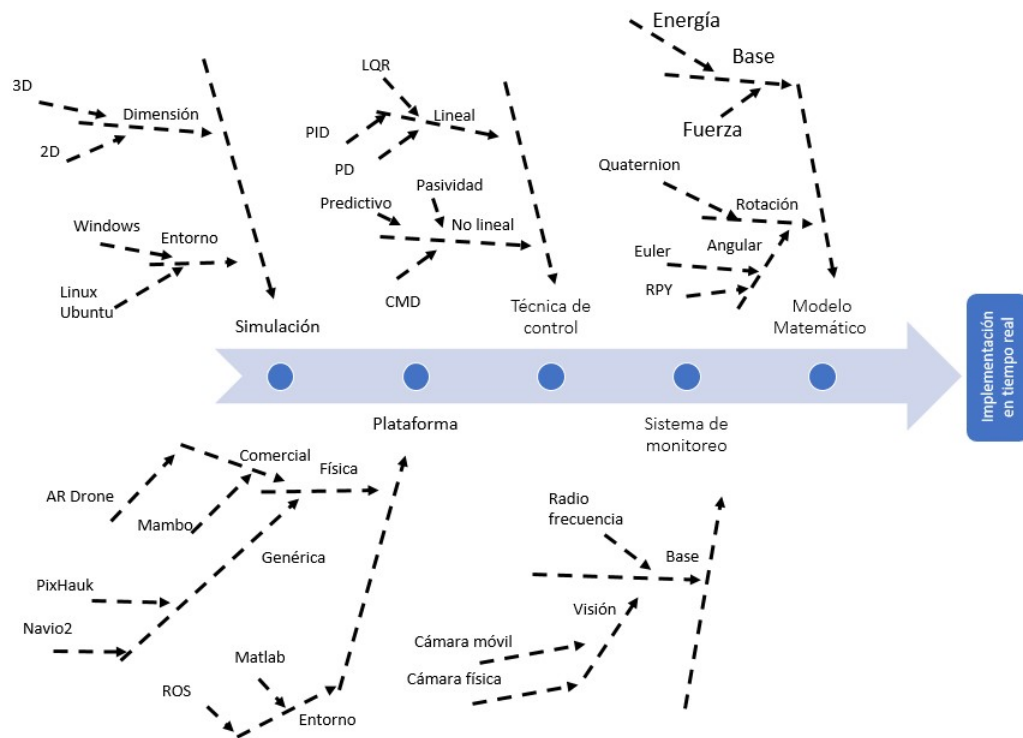


Figura 1.1: Diagrama del pez para el problema de investigación.

Posteriormente se realiza la linealización del modelo dinámico no lineal con base a la serie de Taylor y considerando pequeños desplazamientos angulares. Después, se obtiene un modelo matemático lineal equivalente, se realiza su correspondiente simulación numérica y se compara con la dinámica completa. Además, en dicho periodo, se diseñó un controlador PD y su sintonización para los parámetros físicos del móvil. Después, se realizó una entrega parcial del sistema de monitoreo utilizando el sistema basado en radio frecuencia "GPS indoor", también se entregó una aplicación programada en Matlab que reflejaba los 6 grados de libertad del móvil.

La plataforma de experimentación seleccionada es el vehículo AR Drone 2.0 de la marca Parrot, en el tercer semestre se investigó las características de su

controlador interno en la literatura y documentación oficial, se identificaron y probaron distintas paqueterías para la comunicación y control en Windows. Se seleccionó el controlador por modos deslizantes como una alternativa al PD dada su característica de rechazo de perturbaciones, después, se trabajó en el diseño, caracterización y simulación de ambos controles considerando *a)* los parámetros físicos del móvil, *b)* un trabajo de la literatura de identificación paramétrica. En este mismo periodo se implementó un control PD auxiliado de Matlab en Windows para estabilizar el cuadricóptero en vuelo estacionario, con los resultados experimentales se publicó un artículo en un congreso de investigación estatal.

En ese mismo periodo se trabajó en la implementación de un sistema de monitoreo en base a visión artificial con técnicas de mapeo y localización simultáneas (SLAM) con base al sistema operativo robótico (ROS) en "Linux Ubuntu", después, se implementó dicho sistema considerando *a)* la información dinámica y la transmisión de vídeo que realiza el mismo móvil y *b)* utilizando una Raspberry Pi 3 B+ acondicionada con la IMU MPU9250 y la cámara de Raspberry v1.2.

Por último, sobre final del tercer semestre y en el cuarto semestre, se trabajó en controladores en lazo abierto y cerrado en el entorno de ROS, respecto al lazo cerrado se utilizó el sistema de monitoreo en base a visión para cerrar el lazo de control.

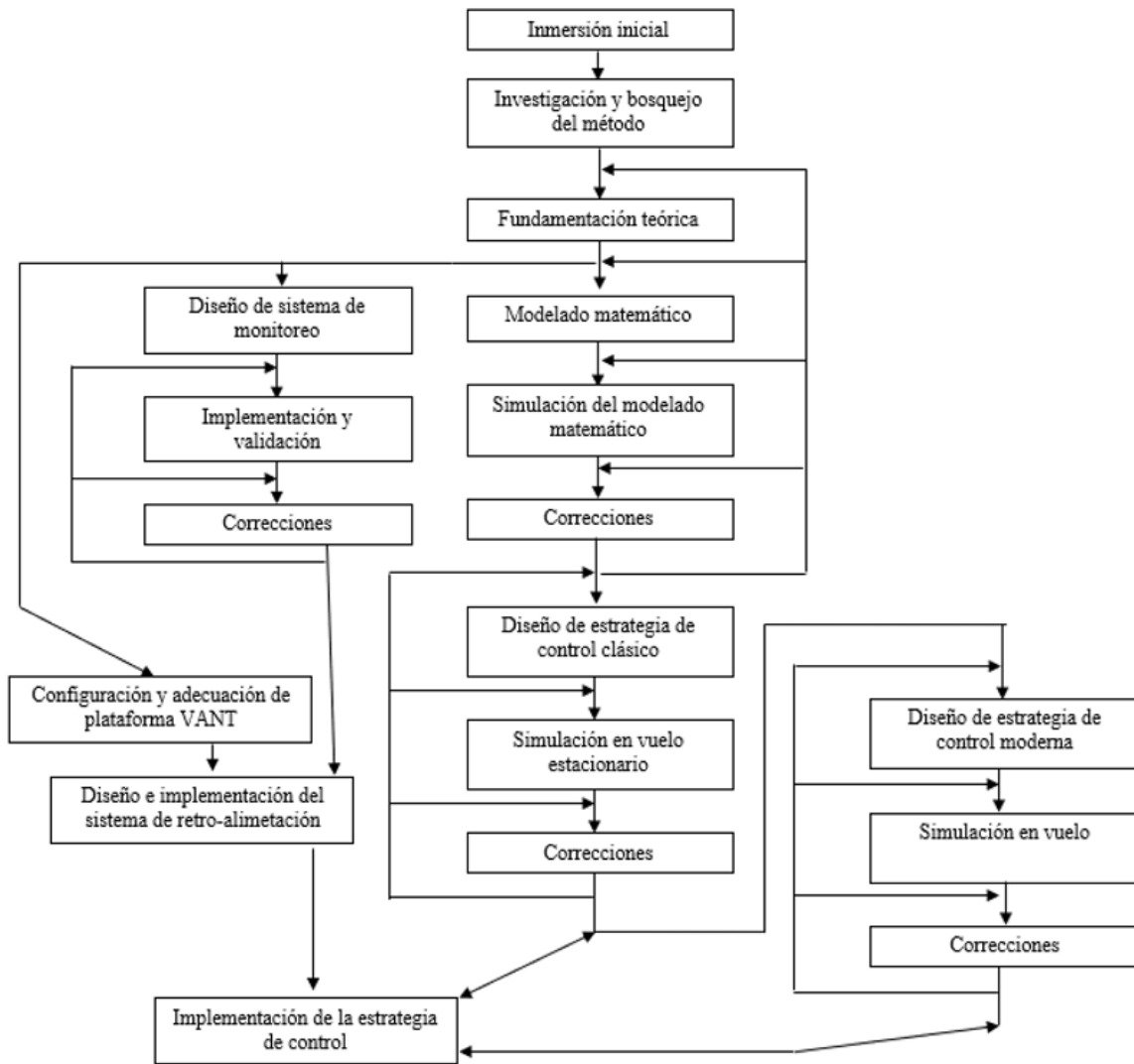


Figura 1.2: Metodología.

Tabla 1.1: Cronograma de actividades.

Actividad	Sem. I	Sem. II	Sem. III	Sem. IV
Inmersión a la problemática.	x			
Diseño y entrega de propuesta de tesis.	x			
Estudio del estado del arte	x	x	x	
Modelo matemático y simulación.		x		
Modelo matemático lineal y simulación.		x		
Diseño y caracterización de controlador PD.		x	x	
Diseño de sistema de monitoreo y retroalimentación.			x	x
Configuración y adecuación de la plataforma de experimentación.		x	x	
Implementación del controlador PD.			x	
Diseño del controlador CMD.			x	
Implementación del controlador CMD.			x	x
Documentación y publicación de tesis.	x	x	x	x
Publicación de artículo de congreso.			x	x
Examen de grado.				x

## 1.8. Organización del trabajo

En esta sección, se proporciona al lector una visión general del contenido de los capítulos que integran esta tesis. Se presentan 5 capítulos, los cuales se detallan a continuación:

- El Capítulo 1: Introducción. Este capítulo es el que proporciona los

antecedentes y la motivación para desarrollar el tema de tesis. Se presenta una revisión del estado del arte de los diferentes temas examinados en esta tesis.

- El Capítulo 2: Vehículo aéreo no tripulado. Este capítulo contiene una síntesis del uso y clasificación de los VANTs, así como la representación del modelo matemático lineal y no lineal de un vehículo tipo cuadrirrotor. También se realiza una descripción del vehículo AR Drone 2.0 de la marca Parrot.
- El Capítulo 3: Estrategias de control y simulación. En forma general, en este capítulo se analiza en detalle el diseño y la aplicación de estrategias de control convencionales y por modos deslizantes para el sistema cuadrirrotor.
- El Capítulo 4: Resultados experimentales. Este capítulo presenta el desarrollo y la aplicación en tiempo real del control por modos deslizantes. También presenta aspectos sobre la plataforma de monitoreo, el uso de visión artificial y el sistema por radiofrecuencia para poder realizar los resultados experimentales.
- El Capítulo 5: Conclusiones. En este capítulo se presentan las conclusiones generales de este trabajo de investigación, se proporciona un breve resumen y discusión de los resultados. Por último se incluyen los trabajos futuros.

## 1.9. Conclusión

Como se reflejó en el estado del arte, el diseño y experimentación de técnicas de control en vehículos aéreos no tripulados es un tópico de investigación que ha estado en auge la última década y media. De manera particular, en la experimentación sobre el cuadrirrotor comercial Parrot AR Drone se exploraron tanto técnicas de control clásico y moderno. Los objetivos presentados toman relevancia significativa en el contexto del ecosistema regional de investigación, ya que reducen la brecha existente entre el estado del arte general y el regional; además, la técnica de visión implementada se encuentra en el límite del estado del arte de la odometría y el SLAM visual.



# CAPÍTULO 2

VEHÍCULOS AÉREOS NO  
TRIPULADOS

# Capítulo 2

## Vehículos aéreos no tripulados

El objetivo principal de este capítulo, es presentar una descripción de los vehículos aéreos no tripulados, así como el modelo matemático dinámico de un cuadrirrotor.

Justo después de brindar la definición formal de un vehículo aéreo o tripulado, en la sección 2.2 se muestran algunas de las distintas formas de clasificar estos vehículos en la literatura. La sección 2.3 brinda una reseña histórica de los vehículos aéreos no tripulados. La sección 2.4 presenta las características principales de la plataforma de experimentación AR Drone 2.0. En la sección 2.5 se desarrolla un modelo matemático basado en las ecuaciones de Newton-Euler y su linealización. Finalmente, la sección 2.6 presenta las conclusiones del capítulo.

### 2.1. Introducción

En el mercado comúnmente se le llama dron a un vehículo aéreo multipropósito generalmente autónomo o controlado a distancia y que carece de operador a bordo. El departamento de defensa de los Estados Unidos los clasifica bajo el nombre de vehículos aéreos no tripulados de la siguiente manera:

*VANT: vehículo aéreo no tripulados que no lleva un operador humano, utiliza fuerzas aerodinámicas para elevar el vehículo, puede volar de manera autónoma o ser pilotado de forma remota, puede ser prescindible o recuperable y puede llevar una carga útil [44].*

Los VANTs son sistemas complejos que generalmente comprenden dos partes, una estación de control y la parte aérea. i) Estación de control: se utiliza por los

operadores del sistema, en ella se asegura la preparación de la misión y la comunicación de datos con la parte aérea y con los organismos que coordinan y controlan el VANT. ii) Parte aérea: está compuesta por una o varias plataformas aéreas dotadas de una carga útil (controladores, sensores, cámaras, etc.) y de un sistema de comunicación.

Los VANTs son vehículos de talla reducida en comparación con los sistemas de vuelo tradicional, son económicamente accesibles, generalmente son más discretos, y su pérdida o destrucción no es tan sensible. Las configuraciones de los VANT son muy variadas, están los basados en rotores (1, 2, 4 o múltiples rotores), los que utilizan alas (fijas y flexibles) y los auto sustentados.

A continuación se enlistan aplicaciones de los VANTs registradas en la literatura, presentadas en dos clasificaciones, uso civil y uso militar[38]. Usos civiles:

- Fotografía aérea.
- Usos agrícolas.
- Fines meteorológicos.
- Monitoreo de tráfico.
- Logística: VANT para operaciones de carga.
- Autoridad policial.
- Servicios de bomberos.
- Servicios de información.
- Investigación y desarrollo: como experimento para futuros estudios.

Usos militares:

- Reconocimiento.
- Vigilancia.
- Objetivo y señuelo: funciona como un misil enemigo o un avión para artillería aérea o terrestre.
- Combate: como avión de combate o bombardero.
- Inteligencia electrónica.

## 2.2. Clasificación de VANTS

La definición citada de VANT con anterioridad no especifica las características físicas del VANT como propulsión, forma de vuelo, despegue y aterrizaje, estructura o chasis mecánico, entre otras, por lo tanto, existen diversas formas de clasificarlo. En [48] se proponen *a)* una clasificación en función de la manera en que realizan la tarea de despegue y aterrizaje, *b)* otra en función de su autonomía y *c)* una clasificación en función de su peso.

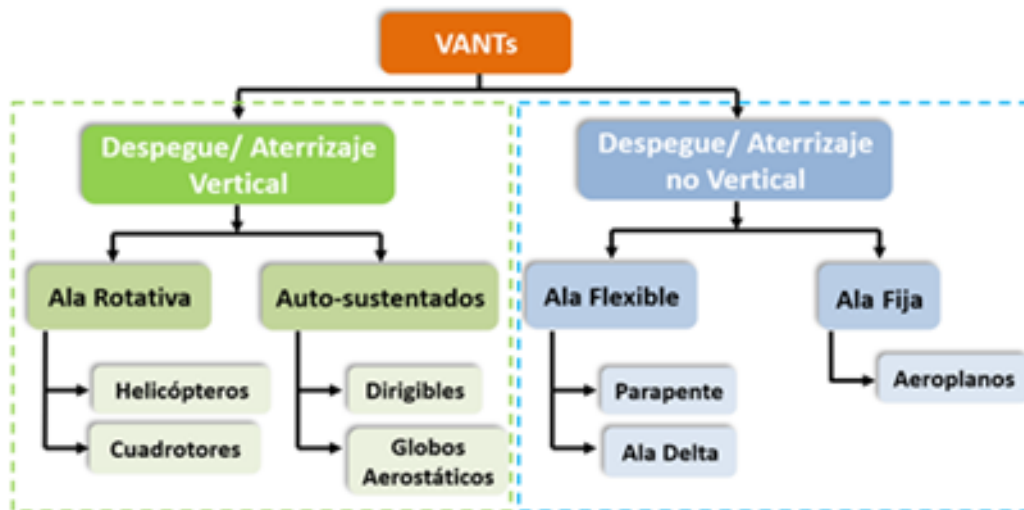


Figura 2.1: Clasificación en función de las tareas despegue y aterrizaje.

Como se observa en la Figura 2.1 los VANTS pueden realizar su tarea de despegue y aterrizaje ya sea de forma vertical, o bien, no vertical. Por ejemplo, dentro de los que realizan estas tareas de manera vertical se encuentran los que utilizan rotores, por ejemplo los helicópteros, cuadrirrotores y hexárotores; en esta sub-clasificación se encuentran también los denominados auto-sustentables, por ejemplo los dirigibles. Por otra parte dentro de los de despegue no vertical se encuentran los de ala fija, que son los más conocidos históricamente, ejemplo de ellos son los aeroplanos; y se encuentran también los de ala flexible, por ejemplo el parapente.

La segunda clasificación es la que está en función de la autonomía del vehículo (véase Tabla 2.1). Los VANTS que se controlan a distancia por un operador tienen una nula autonomía. Si el VANT está programado de manera tal que puede realizar una misión por sí mismo contará con autonomía 1. La autonomía va en aumento en

función de si el vehículo puede cambiar por si mismo de misión, adaptabilidad en tiempo real a fallas y robustez ante éstas, el reconocimiento del entorno, y si estas características las realizan en conjuntos o enjambres.

Tabla 2.1: Clasificación en función de la autonomía.

Nivel	Descripción
0	Vehículo operado remotamente
1	Realiza misión pre-planeada
2	Es capaz de llevar a cabo una misión
3	Respuesta robusta ante fallas/eventos en tiempo real
4	Adaptable ante fallas/eventos en tiempo real
5	Coordinación con otros vehículos en tiempo real
6	Cooperación entre vehículos en tiempo real
7	Conocimiento del espacio de operación
8	Reconocimiento del espacio de operación
9	Reconocimiento del espacio en enjambre
10	Vehículo completamente autónomo

Otra alternativa de clasificación es en función del peso promedio de despegue (PPD). Siendo el PPD el peso de que soporta el VANT al despegar, este está en función de los límites estructurales de su construcción, de las características mecánicas de los actuadores y de la capacidad de almacenaje de las baterías. En la Tabla 2.2 se muestran los nombres de los VANT acorde a esta clasificación.

Tabla 2.2: Clasificación en función del peso promedio de despegue.

Numero	Peso promedio	Nombre
0	Menos de 1 kg	Micro
1	Hasta 1 kg	Mini
2	Hasta 13.5 kg	Pequeño
3	Hasta 242 kg	Ligero
4	Hasta 4332 kg	Normal
5	Mas de 4332	Grande

En función del tipo de misión, se utilizan diferentes motores en los VANT. Estos

incluyen turboventiladores (ejemplo Global Hawk, uso militar), motores de dos tiempos, pistón (Predator, uso militar), rotativos, turbohélice, eléctricos (Dragon Eye, uso militar) y de hélice.

Los VANTs con inteligencia computacional se pueden clasificar en función de las características necesarias para su misión, tales como resistencia, alcance, peso y altitud máxima:

### **2.3. Reseña histórica.**

Vehículo aéreo no tripulado es un término que se generalizó a principios de la década de 1990 y reemplazó la expresión previamente preferida de vehículo pilotado a distancia. Aunque VANT es un concepto reciente que se ha vuelto popular debido a usos especialmente en las operaciones de la OTAN durante la Guerra de Kosovo y operaciones estadounidenses como la operación libertad duradera [38], tienen una historia casi tan antigua como los aviones tripulados de la primera guerra mundial.

Unos años después del primer vuelo en avión tripulado, el Dr. Cooper y Elmer Sperry diseñaron el estabilizador giroscópico automático, mediante el cual mantenían un avión volando recto y nivelado. Esta tecnología se utilizó para convertir un avión de entrenamiento Curtiss N-9 de la Armada de los E.U.A. en el primer vehículo pilotado a distancia. Los primeros vehículos pilotados a distancia se probaron en E.U.A. durante la primera guerra mundial, mas no se implementaron en combate, ya en la segunda guerra mundial, Alemania se aventajó y demostró su utilidad en combate. Después de las dos guerras, los militares reconocieron su potencial combate e iniciaron programas de desarrollo que llevaron, pocas décadas después, a sistemas sofisticados, especialmente en E.U.A. e Israel, como el Predator (General Atomics) o el Pioneer (PUAV)[39].

Históricamente, el Kettering Bug construido por Charles Kettering para la marina de E.U.A. en 1918 puede ser aceptado como el primer VANT. Este prototipo fue lo suficientemente exitoso como para hacer que el ejército pidiera otros adicionales, pero nunca voló operativamente debido al fin de la primera guerra mundial. Por otro lado, es posible mostrar el error del relámpago, que fue utilizado sustancialmente por los E.U.A. durante la guerra de Vietnam entre 1964 y 1972, como el antepasado de los vehículos aéreos no tripulados modernos.

El primer helicóptero no tripulado fue el construido por Forlanini en 1877, no

estaba estabilizado activamente ni era orientable. Con los avances tecnológicos sobresalientes después de la segunda guerra mundial, fue posible construir y controlar helicópteros no tripulados. La empresa Gyrodyne of America inició el famoso programa DASH para la marina.

El mercado militar de helicópteros no tripulados se hizo evidente. Se desplegó un intenso esfuerzo de investigación y se lograron resultados como el A160 Hummingbird, un helicóptero de larga duración capaz de volar 24 hrs dentro de un alcance de 3,150 km. El campo de batalla del futurística perteneció al Rotorcraft armado de combate no tripulado, pero este VANT quedó solamente como prototipo.

## 2.4. Parrot AR. Drone.

El Parrot AR. Drone basa su diseño en un cuadrirrotor clásico, tal como se muestra en la Figura 2.2. Se utilizan 4 motores sin escobillas para alimentar las 4 hélices, que en conjunto, constituyen generadores de empuje variable. Cada motor está montado sobre bases de plástico PA66 de alta resistencia, en las cuales se encuentra montado el circuito de control de los motores, cada controlador de motor utiliza su propio microcontrolador ATMEGA8L de 8 bits y sistema de corte para apagar el motor en caso de que un obstáculo se interponga en la trayectoria de la hélice.

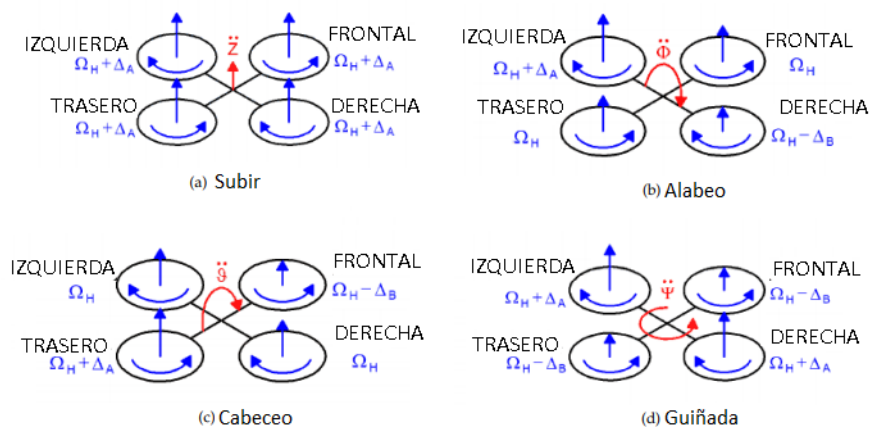


Figura 2.2: Movimientos del AR Drone.

Los 4 generadores de empuje se unen mediante 4 tubos de fibra de carbono y una cruz central reforzada con fibra de plástico. La estructura de carbono soporta

una estructura de polipropileno expandido que lleva la batería de polímero de litio en una estructura de plástico. Se pueden montar dos carcasas protectoras al cuadricóptero (Figura 2.3), la primera es un cuerpo de cubierta para vuelo en interiores que cuenta con protección para las hélices, con ella el peso del móvil es 480 gr. La segunda es la carcasa para vuelos exteriores sin la protección, con ella el peso es 380 grs.



Carcasa para vuelo interior.

Carcasa para vuelo exterior.

Figura 2.3: Carcasas protectoras Parrot AR Drone.

A continuación, se alistan las características de hardware del Parrot AR Drone:

- CPU (ARM Cortex A8, OMAP 3630) a 1 GHz.
- GPU (PowerVR SGX530) a 800 MHz, 1 GB.
- RAM, 128 MB de ROM.
- 2 cámaras web: Cámara vertical 63fov @ 60fps, 240p para mediciones de velocidad horizontal. Cámara frontal 93fov @ 30fps, 720p.
- 1 WiFi.
- 1 ultrasónico alcance máximo de 6 m.
- 1 barómetro ( $\pm 10$ pa).
- 1 IMU de 9 DOF (contiene acelerómetro de 3 ejes ( $\pm 50$  mg), giroscopio de 2 ejes (2,000 grados / seg), giroscopio de guiñada preciso (XB3500CV, deriva 12 grados / min dinámico, 4 grados / min estático). Magnetómetro de 3 ejes ( $\pm 6$  grados).



- 1 puerto USB (GPS y módem LTE).
- 4 motores sin escobillas a 28,500 RPM, 14.5 W, relación de engranajes 1: 8.75, con tablero de control (ATMEGA8L).
- Hasta 5 m /seg, 13 minutos de vuelo continuo.
- Baterías LiPo de 1,000 mA, 11.1 V (capacidad de descarga 15C, 80 gramos) el voltaje disminuye de carga completa (12.5 voltios) a carga baja (9 voltios).

La empresa Parrot creó un kit de desarrollo de software (SDK), con base en Linux. En mayo de 2012 la empresa publica “ArDrone: the flying video game. Developer Guide, SDK 2.0”, documento que brinda una guía para el desarrollo de aplicaciones basadas en AR Drone para la versión 1.0 y 2.0 [42].

El control del AR Drone se realiza a través de 3 canales de comunicación. El control y la configuración se efectúa enviando comandos AT por el puerto UDP 5556. La información acerca del móvil, tal estado, posición, velocidad y velocidad de rotación de los motores, se envía por el puerto UDP 5554. La transmisión de vídeo se realiza a través del puerto UDP 5555.

Dentro del SDK se encuentra la librería AR Drone, a la cual se le subordinan los comandos AT y la transmisión de datos. Según las pruebas realizadas por los desarrolladores la transmisión de los comandos debe ser como máximo cada 30 ms para lograr un control adecuado. La Tabla 2.3 muestra un resumen de los comandos AT disponibles.

Tabla 2.3: Comandos AT en SDK.

Comando	Argumentos	Micro
AT*PCMD	flag, roll, pitch, gaz, yaw	Mover el vehículo
AT*PCMD-MAG	flag, roll, pitch, gaz, yaw, psi, psi-accurary	Mover el vehículo de manera absoluta
AT*FTRIM		Establecer referencia horizontal
AT*CONFIG	key, value	Configuración
AT*CONFIG-IDS	session, user, aplicacion ids	indicador de comando de configuración
AT*COMWDG		Restablecer comunicación
AT*CALIB	device number	Calibración magnetómetro
AT*REF	input	Despegue, aterrizaje, emergencia.

Los comandos para mover el cuadricóptero son *AT \* PCMD* y *AT \* PCMD – MAG*, este último está disponible solamente en iOS. A continuación, se describe el comando *AT\*PCMD*, comando que hace que el móvil se mueva solamente cuando el vehículo se encuentra en el aire.

Argumentos del *AT \* PCMD*:

- flotante phi: ángulo izquierdo / derecho [-1 a 1], un ángulo de flexión izquierda / derecha, con 0 como plano horizontal y valores negativos de flexión hacia la izquierda.
- flotante theta: ángulo frontal / posterior [-1 a 1], un ángulo de flexión frontal / posterior, siendo 0 el plano horizontal y valores negativos de flexión hacia adelante.
- flotante gaz: velocidad vertical [-1 a 1], una velocidad vertical.
- flotante psi: velocidad angular [-1 a 1], velocidad angular alrededor del eje de guiñada.

Los argumentos de esta función no son directamente los valores de los parámetros de control, sino un porcentaje de los valores máximos correspondientes establecidos

en los parámetros del móvil. Todos los parámetros deben ser valores flotantes entre -1 y 1.

## 2.5. Modelo matemático.

En esta sección se presenta el conjunto de ecuaciones diferenciales que representan la dinámica de un vehículo tipo cuadrirrotor bajo la formulación Newton-Euler.

### 2.5.1 Descripciones geométricas.

El cuadrirrotor está caracterizado por contar con 4 rotores anclados simétricamente en los extremos del cuerpo rígido en forma de cruz [48]. El eje de rotación de los motores es perpendicular al plano formado por el chasis, en consecuencia, realiza de forma vertical sus tareas de despegue y aterrizaje. En la Figura 2.4 se muestra un bosquejo representativo de un cuadrirrotor.

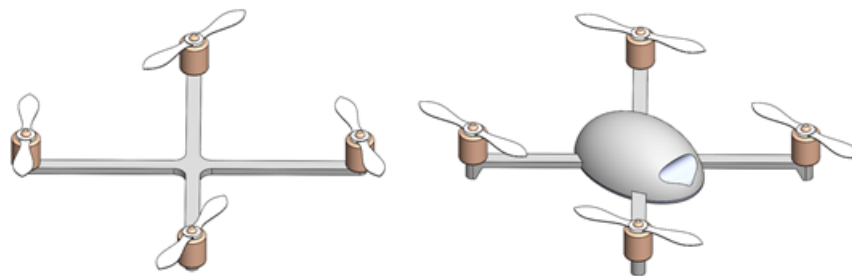


Figura 2.4: Bosquejo de un cuadrirrotor.

Los VANTs se clasifican como robots de base móvil o simplemente robots móviles, ya que ni un eslabón del robot se encuentra anclado o fijo y el marco de referencia se encuentra en movimiento junto al robot. Debido a lo anterior se acostumbra asociarles al menos dos tramas distintas, una inercial o fija, por ejemplo, en una estación de monitoreo, y al menos una en el cuerpo del robot. Los sistemas de coordenadas que describen la posición se seleccionan a conveniencia del investigador, en el presente trabajo se usa un sistema coordenadas rectangulares, y por convención se escoge una trama móvil localizada en el centro de masa del cuadrirrotor.

Las coordenadas referenciadas al marco inercial se considerarán como absolutas y las asociadas a los marcos de referencias móviles se consideran coordenadas locales. La orientación de los ejes se obtiene con la regla de la mano derecha, se considera positivo un desplazamiento angular en sentido anti-horario tal como se muestra en la Figura 2.5, la orientación y la posición absolutas se obtienen mediante la matriz de rotación  $R(\eta)$  parametrizada con los ángulos de Euler y el vector de posición.

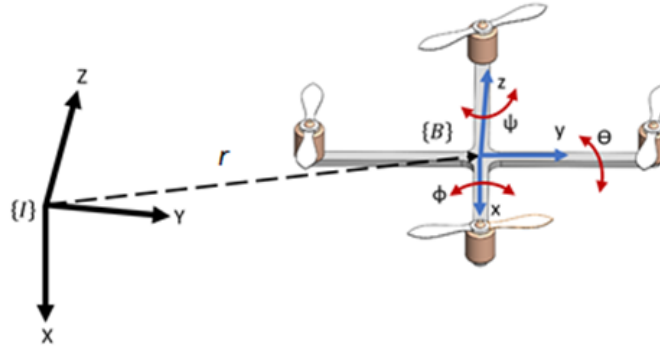


Figura 2.5: Ángulos de Euler.

**Ángulos de Euler Z-Y-X.** Para recordar, la descripción de la orientación por los ángulos de Euler se obtiene mediante el siguiente procedimiento:

1. Cuando ambas tramas coinciden gire  $\{B\}$  sobre  $Z_B$  en un ángulo  $\psi$ .
2. Gire  $Y_B$  en un ángulo  $\theta$ .
3. Gire  $X_B$  en un ángulo  $\phi$ .

Cada una de estas rotaciones se lleva a cabo sobre un eje del sistema  $\{B\}$ , en vez del sistema de referencia fija  $\{A\}$ .

Dado  $\eta = [\phi, \theta, \psi]^T \in R^3$  donde  $\phi, \theta, \psi$  son los ángulos de rotación del cuerpo alrededor de los ejes  $x$ ,  $y$  y  $z$  respectivamente, la matriz de rotación  $R(\eta) = \{B\} \rightarrow \{I\}$  se describe como el producto sucesivo de las rotaciones alrededor de los ejes principales en el orden  $z, x, y$   $R(\eta) = [R_{\psi,z}][R_{\phi,x'}][R_{\theta,y''}]$ ,

$$R(\eta) = \begin{bmatrix} c_\psi & -s_\psi & 0 \\ s_\psi & c_\psi & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & c_\phi & -s_\phi \\ 0 & s_\phi & c_\phi \end{bmatrix} \begin{bmatrix} c_\theta & 0 & s_\theta \\ 1 & 0 & 0 \\ -s_\theta & 0 & c_\theta \end{bmatrix}, \quad (2.1)$$

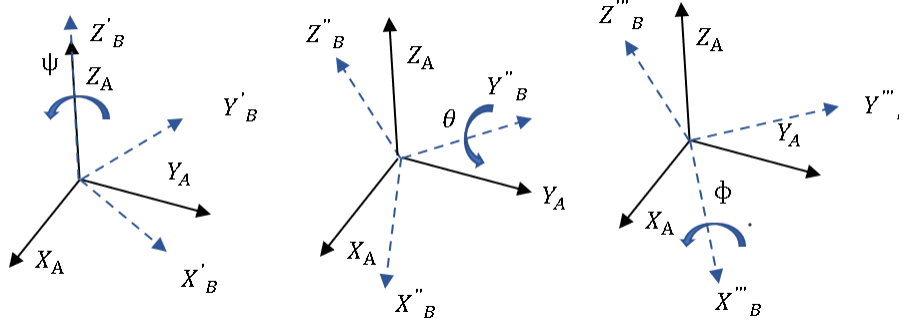


Figura 2.6: Ángulos de Euler.

$$R(\eta) = \begin{bmatrix} c_\psi c_\theta - s_\psi s_\theta s_\phi & -c_\phi s_\psi & c_\psi s_\theta + c_\theta s_\phi s_\psi \\ c_\theta s_\psi + c_\psi s_\theta s_\phi & c_\phi c_\psi & s_\psi s_\theta - c_\theta s_\phi c_\psi \\ -c_\phi s_\theta & s_\phi & c_\phi c_\theta \end{bmatrix}. \quad (2.2)$$

Usualmente en robótica a los ángulos  $\phi, \theta, \psi$  se les llama ángulo de balanceo (roll), ángulo de inclinación (pitch) y ángulo de orientación (yaw) respectivamente y la matriz de rotación mostrada en la Ecuación 2.1 se abrevia con las siglas RPY (o YRP debido a la sucesión en la multiplicación).

Para expresar la posición de un vector  $V_B$  definido en la trama  $\{B\}$ , respecto a una trama distinta  $\{I\}$  es necesario sumarle el vector  $p_{IB}$  al vector  $V_B$ , donde  $p_{IB}$  representa el desplazamiento existente entre los orígenes de las tramas, se describe con operación  $V_I = V_B + p_{IB}$ . Ahora bien, si se desea expresar la posición de  $V_B$  en  $\{B\}$  respecto a  $\{I\}$  cuando  $V_B$  sufre una rotación, primero es necesario aplicar la rotación  $R(\eta)$  a  $V_B$  y posterior a eso sumar el vector de posición  $p_{IB}$ .

Se considera que el semi eje positivo  $x$  de la trama  $\{B\}$  apunta hacia el frente del cuadrirrotor, y a partir de él se enumeraran los rotores en sentido contrario a las manecillas del reloj como se muestra en la Figura 2.7, los rotores 1 y 3 giran en sentido horario, mientras que 2 y 4 en sentido antihorario.

El cuadrirrotor describe un movimiento positivo de alabeo disminuyendo la velocidad angular del rotor 4 y aumentándola en el rotor 2, mientras que 1 y 3 mantienen velocidad constante, o bien  $\omega_4 < \omega_2$  y  $\omega_1 = \omega_3$ ; cuando  $\omega_4 > \omega_2$  y  $\omega_1 = \omega_3$  el cuadrirrotor describirá un movimiento de alabeo negativo. De manera similar, se produce un movimiento de cabeceo cuando  $\omega_3 < \omega_1$  o  $\omega_3 > \omega_1$  y  $\omega_2 = \omega_4$ . Por último, el movimiento de guiñada se obtiene aumentando  $\omega_1$  y  $\omega_3$  mientras que se

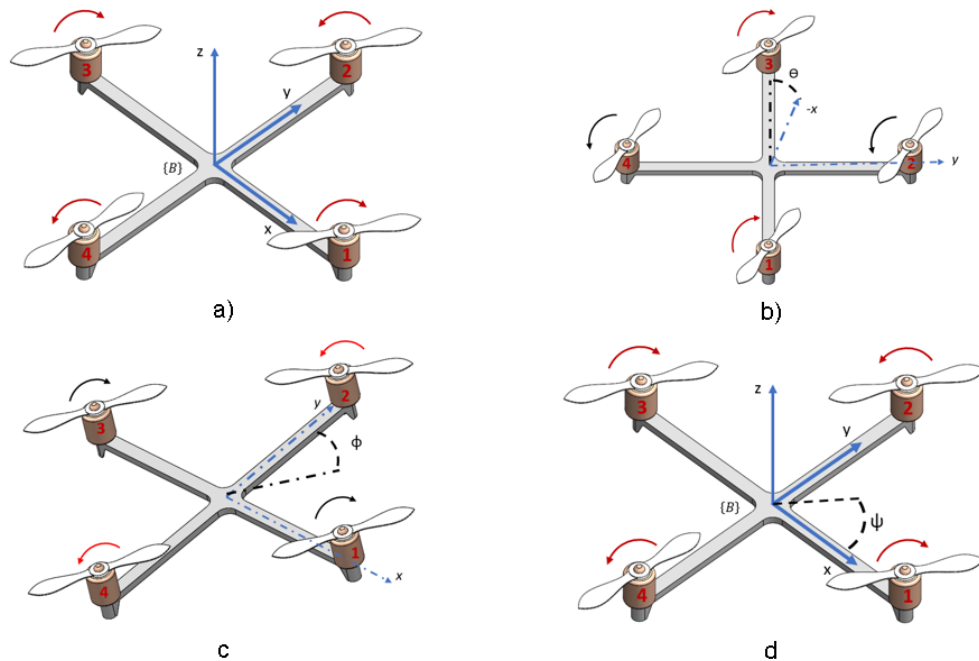


Figura 2.7: Movimientos en un cuadrirrotor.

disminuyen  $\omega_2$  y  $\omega_4$ , y viceversa.

Si lo que se desea es lograr una elevación en  $z$  el empuje total combinado de los 4s rotores en dirección del eje debe superar a la fuerza de gravedad. La fuerza de empuje individual se obtiene con  $u_i = k_F \omega_i^2$ , donde  $k_F$  es la constante de empuje y  $\omega_i$  es la velocidad angular del  $i$ -ésimo rotor.

### 2.5.2 Modelo matemático no lineal.

Tal como se mencionó previamente 4 de los 6 movimientos independientes del cuadrirrotor se pueden controlar de forma directa, esto se debe a que el cuadrirrotor es un sistema sub-actuado, en concreto consta de 6 grados de libertad y 4 entradas.

La dinámica del cuadrirrotor se puede considerar un movimiento complejo compuesto por la dinámica traslacional referenciada al marco inercial  $I$  y a una rotación centroidal alrededor del centro de masa del mismo. Considerando el cuadrirrotor como un cuerpo rígido, dado su chasis en forma en cruz y la simetría con la que se posicionan los actuadores sobre éste, su centro de masa está situado en la intersección de la cruz, lugar donde se le adjunta el marco de referencia  $\{B\}$ .

Sea  $r = p_{IB} = [x, y, z]^T$  el vector de posición que relaciona el origen del marco de referencia  $\{B\}$  con  $\{I\}$ , véase Figura 2.8, mismo que representa la posición relativa del cuadricóptero respecto a la trama inercial  $I$ , y sean  $v = \dot{r}$  y  $a = \ddot{r}$  la primera y segunda derivada de  $r$  con respecto a el tiempo  $t$  respectivamente, acorde a la segunda ley de Newton, la fuerza total  $F$  aplicada sobre el cuadricóptero se obtiene por  $F = ma = m\ddot{r}$ , donde  $m$  representa la masa del móvil.

De esta manera, la ecuación de equilibrio dinámico translacional del centro de masa del cuadricóptero referenciada a  $\{I\}$  se expresa como sigue:

$$m\ddot{r} = \begin{pmatrix} 0 \\ 0 \\ -mg \end{pmatrix} + R(\eta) \begin{pmatrix} 0 \\ 0 \\ u \end{pmatrix}, \quad (2.3)$$

donde  $R(\eta)$  representa la orientación relativa a  $\{I\}$  mediante los ángulos de Euler (Ecuación 2.1),  $\eta = (\phi, \theta, \psi)^T$ , y  $u$  es la fuerza total de empuje en dirección de  $z$  en el marco  $\{B\}$ , obtenida por

$$u = \sum_{i=1}^4 u_i = k_F \sum_{i=1}^4 \omega_i^2, \quad (2.4)$$

$k_F$  es una constante y  $\omega_i$  es la velocidad angular del  $i$ -ésimo rotor. La dinámica translacional del cuadricóptero utilizando la notación corta  $s_\theta = \sin\theta$  y  $c_\theta = \cos\theta$  se expresa de la siguiente manera:

$$m\ddot{r} = \begin{bmatrix} 0 \\ 0 \\ -mg \end{bmatrix} + \begin{bmatrix} c_\psi s_\theta + c_\theta s_\phi s_\psi \\ s_\psi s_\theta - c_\theta s_\phi c_\psi \\ c_\theta c_\phi \end{bmatrix} u, \quad (2.5)$$

Considérese ahora el tensor de inercia  $I \in R^{3 \times 3}$  como un indicador de un cuerpo para almacenar energía cinética rotacional  $E_c = \frac{1}{2} I \varepsilon^2$ , donde  $\varepsilon$  representa la velocidad angular de  $\{B\}$  respecto a  $\{I\}$  en  $\{B\}$ , y como un indicador del par de torsión necesario para acelerar un cuerpo rotacionalmente  $H = I\varepsilon$ .

$$I = \begin{bmatrix} I_{xx} & I_{xy} & I_{xz} \\ I_{yx} & I_{yy} & I_{yz} \\ I_{zy} & I_{zy} & I_{zz} \end{bmatrix}. \quad (2.6)$$

Compuesto por los momentos principales  $I_{xx}$ ,  $I_{yy}$  e  $I_{zz}$ .  $I_{xx} = \int (y^2 + z^2) dm$ ,  $I_{yy} = \int (x^2 + z^2) dm$  y  $I_{zz} = \int (y^2 + x^2) dm$ . Y los productos de inercia  $I_{xy}$ ,  $I_{xz}$ ,  $I_{yz}$   $I_{xy} = I_{yx} = - \int xy dm$ ,  $I_{xz} = I_{zx} = - \int xz dm$ ,  $I_{yz} = I_{zy} = - \int zy dm$ .

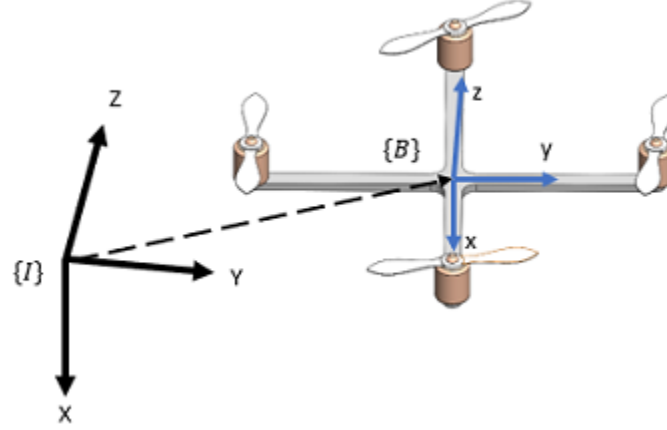


Figura 2.8: Sistemas de referencia.

Dada la distribución simétrica del móvil respecto a la trama  $\{B\}$ , considerando una densidad uniforme en el material de construcción los productos de inercia son idénticos a cero, de esta manera  $I$  se reduce a

$$I = \begin{bmatrix} I_{xx} & 0 & 0 \\ 0 & I_{yy} & 0 \\ 0 & 0 & I_{zz} \end{bmatrix}. \quad (2.7)$$

El momento resultante alrededor del centro de masa generado por fuerzas externas es igual a la razón de cambio de la cantidad de movimiento angular alrededor del mismo,  $\tau_0 = \dot{H} = \frac{d}{dt}I\varepsilon$ , donde  $H$  es el momento angular y  $\varepsilon$  la velocidad angular en  $\{B\}$  con respecto a  $\{I\}$  y  $\bar{\tau}_0 = [\tau_2, \tau_3, \tau_4]^T \in R^3$ , siendo estas cantidades vectoriales la razón de cambio se describe por

$$\bar{\tau}_0 = \dot{H} = I\dot{\varepsilon} + \varepsilon \times I\varepsilon. \quad (2.8)$$

El par generado por una fuerza se calcula mediante el producto cruz de la fuerza y la distancia al centro de rotación  $\tau = F \times d$ , debido a la posición de los actuadores la fuerza generada por estos es ortogonal respecto al plano  $XY$  del marco  $\{B\}$  con lo que es posible expresar la multiplicación sin hacer uso del operador  $\times$ . El par  $M_i$  de reacción alrededor del centro de masa producido por el  $i$ -ésimo rotor con dirección opuesta a la rotación de las hélices se calcula con  $M_i = k_M \omega_i^2$ , donde  $k_M [kgm^2]$  es el coeficiente de arrastre, el cual depende de la aceleración del  $i$ -ésimo rotor multiplicada por el momento de inercia del mismo [48].



Por el principio de d'Alembert, el movimiento de un cuerpo rígido queda completamente definido por la resultante de las fuerzas externas y el momento resultante alrededor del centro de masa del cuerpo, además es posible trasladar una fuerza a través de su línea de acción. De esta manera el par generado por los actuadores combinados queda definido por:

$$\tau_2 = L [u_2 - u_4], \quad (2.9)$$

$$\tau_3 = L [u_3 - u_1], \quad (2.10)$$

$$\tau_4 = M_1 + M_3 - M_2 - M_4 \quad (2.11)$$

donde  $L$  es la distancia del rotor al centro de masa del móvil,  $\tau_2$  representa la diferencia de par de los rotores que giran en sentido antihorario,  $\tau_3$  la diferencia de par de los que giran en sentido horario y  $\tau_4$  la diferencia del par de reacción de los 4 rotores.

$$u_T = \begin{bmatrix} u \\ \tau_2 \\ \tau_3 \\ \tau_4 \end{bmatrix} = \begin{bmatrix} k_F & k_F & k_F & k_F \\ 0 & Lk_F & 0 & -Lk_F \\ -Lk_F & 0 & Lk_F & 0 \\ k_M & -k_M & k_M & -k_M \end{bmatrix} \begin{bmatrix} \omega_1^2 \\ \omega_2^2 \\ \omega_3^2 \\ \omega_4^2 \end{bmatrix} \quad (2.12)$$

Con  $k_F$ ,  $k_M$  y  $L$  distintas de cero, la matriz anterior es invertible y es posible despejar a favor de las velocidades angulares  $\omega_i$  considerando a  $u$  y a  $\bar{\tau}_0$  como las entradas de control.

De la Ecuación 2.8 se puede despejar a favor del producto  $I\dot{\varepsilon}$  como sigue

$$I\dot{\varepsilon} = \bar{\tau}_0 - \varepsilon \times I\varepsilon, \quad (2.13)$$

$$I \begin{bmatrix} \dot{p} \\ \dot{q} \\ \dot{r} \end{bmatrix} = \begin{bmatrix} L[u_2 - u_4] \\ L[u_3 - u_1] \\ M_1 + M_3 - M_2 - M_4 \end{bmatrix} - \begin{bmatrix} p \\ q \\ r \end{bmatrix} \times I \begin{bmatrix} p \\ q \\ r \end{bmatrix}, \quad (2.14)$$

$$I \begin{bmatrix} \dot{p} \\ \dot{q} \\ \dot{r} \end{bmatrix} = \begin{bmatrix} L[u_2 - u_4] \\ L[u_3 - u_1] \\ M_1 + M_3 - M_2 - M_4 \end{bmatrix} - \begin{bmatrix} qr [I_{zz} - I_{yy}] \\ pr [I_{zz} - I_{xx}] \\ -pq [I_{xx} - I_{yy}] \end{bmatrix}, \quad (2.15)$$

$$I\dot{\varepsilon} = \begin{bmatrix} \tau_2 \\ \tau_3 \\ \tau_4 \end{bmatrix} + \begin{bmatrix} -qr [I_{zz} - I_{yy}] \\ -pr [I_{zz} - I_{xx}] \\ pq [I_{xx} - I_{yy}] \end{bmatrix}. \quad (2.16)$$

Donde  $\varepsilon$  es el vector velocidad en  $\{B\}$  y esta compuesto por  $\varepsilon = [p, q, r]^T$ . Es posible encontrar una matriz de transformación  $W$ , mediante la cual se relacione el vector  $\varepsilon$  con la derivada temporal de los ángulos de Euler  $\eta = [\dot{\theta}, \dot{\phi}, \dot{\psi}]^T$  expresados en  $\{I\}$ , dados los vectores unitarios  $i = [1, 0, 0]^T$ ,  $j = [0, 1, 0]^T$  y  $k = [0, 0, 1]^T$ ,

$$\varepsilon = \dot{\theta}j + \dot{\phi}[R_{y,\theta}]^T i + \dot{\psi}[R_{y,\theta}]^T [R_{x,\phi}]^T k, \quad (2.17)$$

$$\varepsilon = W\dot{\eta} \quad (2.18)$$

donde

$$W = \begin{pmatrix} c_\theta & 0 & -c_\phi s_\theta \\ 0 & 1 & s_\phi \\ s_\theta & 0 & c_\theta c_\phi \end{pmatrix}. \quad (2.19)$$

Aplicando la regla de la cadena de derivación en 2.18 se tiene  $I \frac{d}{dt} \varepsilon = I \frac{d}{dt} (W\dot{\eta}) = I(\dot{W}\dot{\eta} + W\ddot{\eta})$  y sustituyendo en 2.13 y despejando se obtiene

$$IW\ddot{\eta} = \bar{\tau}_0 - W\dot{\eta} \times IW\dot{\eta} - I\dot{W}\dot{\eta} \quad (2.20)$$

$$I_r \ddot{\eta} = \bar{\tau}_0 + \bar{C} \quad (2.21)$$

La ecuación 2.21 representa la dinámica rotacional en  $\{I\}$

$$\bar{C} = [C_1, C_2, C_3]^T, \quad (2.22)$$

$$I_r = \begin{pmatrix} I_{xx}c_\theta & 0 & -I_{xx}c_\phi s_\theta \\ 0 & I_{yy} & I_{yy}s_\phi \\ I_{zz}s_\theta & 0 & I_{zz}c_\theta c_\phi \end{pmatrix}, \quad (2.23)$$

donde  $C_1 = [I_{xx} + I_{yy} - I_{zz}]\dot{\phi}\dot{\theta}s_\theta + [-I_{xx} + I_{yy} - I_{zz}]\dot{\phi}\dot{\psi}s_\theta s_\phi + [I_{xx} + I_{yy} - I_{zz}]\dot{\phi}\dot{\psi}c_\theta c_\phi + [I_{yy} - I_{zz}]\dot{\psi}^2 s_\phi c_\phi c_\theta$ ,

$$C_2 = [-I_{yy} + [-I_{xx} + I_{zz}]c_{2\theta}]\dot{\phi}\dot{\psi}c_\phi + [-I_{xx} + I_{zz}][\dot{\phi}^2 - \dot{\psi}^2 c_\phi^2]c_\theta s_\theta \text{ y}$$

$$C_3 = [I_{xx} - I_{yy} - I_{zz}]\dot{\phi}\dot{\theta}c_\theta + [I_{xx} - I_{yy} + I_{zz}]\dot{\phi}\dot{\psi}s_\phi c_\theta + [-I_{xx} + I_{yy} + I_{zz}]\dot{\phi}\dot{\psi}s_\theta c_\phi + [-I_{xx} + I_{yy}]\dot{\psi}^2 s_\phi c_\phi s_\theta.$$

### 2.5.3 Modelo matemático lineal.

Ahora se procede a obtener la linealización del modelo asumiendo que el movimiento del cuadrirrotor será controlado de manera tal que los desplazamientos angulares serán próximos a cero. Es decir  $\phi = \theta \approx 0$  con lo que las operaciones

trigonométricas  $\cos(\phi) = \cos(\theta) \approx \cos(0) \approx 1$  y  $\sin(\phi) = \sin(\theta) \approx \sin(0) \approx \phi \approx \theta \approx 0$ , y  $\psi = \psi_d$ , sustituyendo estos valores en  $I_r$  se reduce a  $I$ ,  $W$  y  $R(\eta)$ , estas matrices se reducen a la identidad y el vector  $\bar{C}$  se vuelve cero. De esta manera se tiene la representación lineal de la dinámica translacional:

$$\ddot{r} = \begin{pmatrix} 0 \\ 0 \\ -g \end{pmatrix} + \begin{pmatrix} \theta c_{\psi_d} + \phi s_{\psi_d} \\ \theta s_{\psi_d} - \phi c_{\psi_d} \\ 1 \end{pmatrix} \frac{u}{m}, \quad (2.24)$$

y de la dinámica rotacional:

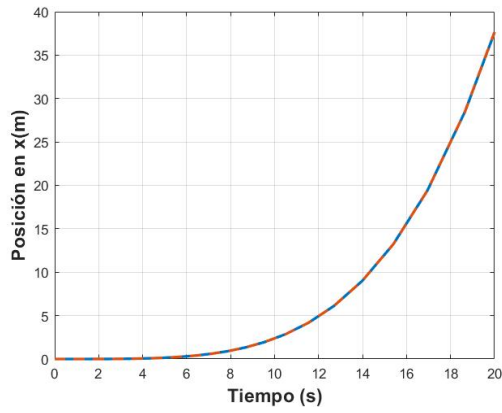
$$\ddot{\eta} = \begin{pmatrix} \frac{\tau_2}{I_{xx}} \\ \frac{\tau_3}{I_{yy}} \\ \frac{\tau_4}{I_{zz}} \end{pmatrix} \quad (2.25)$$

Para efectos de simulación se consideran los valores mostrados en la Tabla 2.4. La Figura 2.9 muestra la dinámica del modelo no lineal y del modelo linealizado, en azul la dinámica completa no lineal y en naranja con línea no continua la dinámica linealizada, ambos para  $u = 3.9743N$ ,  $\tau_2 = \tau_3 = -1.0064 \times 10^{-4}Nm$  y  $\tau_4 = 0Nm$ .

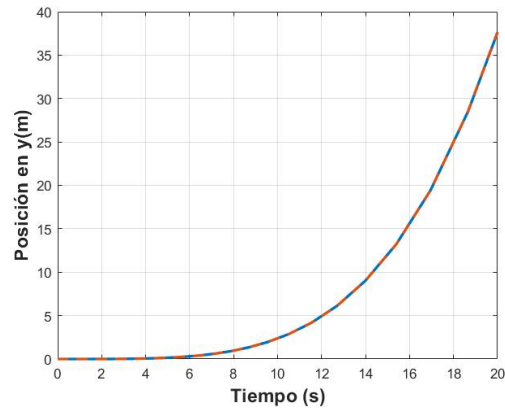
Tabla 2.4: Tabla de parámetros de simulación.

Parámetros	Valor
$m$	$0.4 \text{ kg}$
$I_{xx} = I_{yy} = 2I_{zz}$	$0.177 \frac{\text{kg}}{\text{m}^2}$
$L$	$0.1 \text{ m}$
$k_F = k_M$	$0.1 \frac{\text{kg}}{\text{m}}$

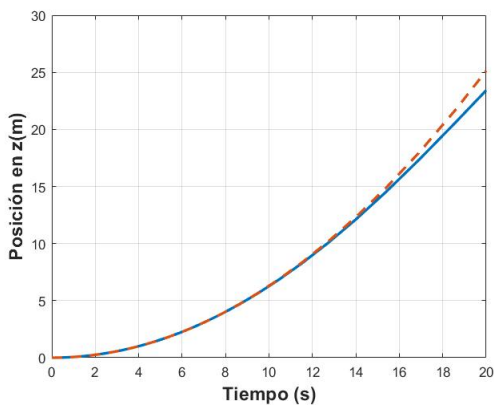
En la Figura 2.9 se observa la evolución de la dinámica de cada uno de los estados del sistema cuadrirrotor, es fácil visualizar que estas dinámicas en lazo abierto tanto del modelo no lineal y del linealizado en cierta región de equilibrio son muy similares.



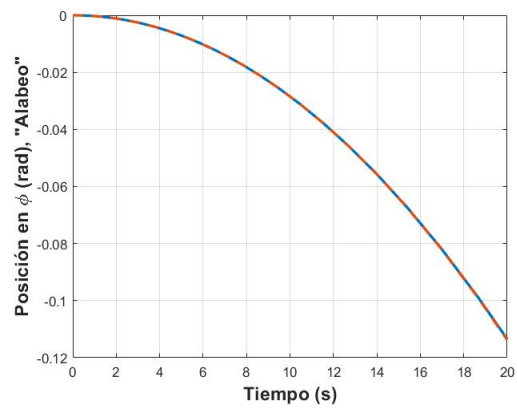
Posición en  $x$  (m).



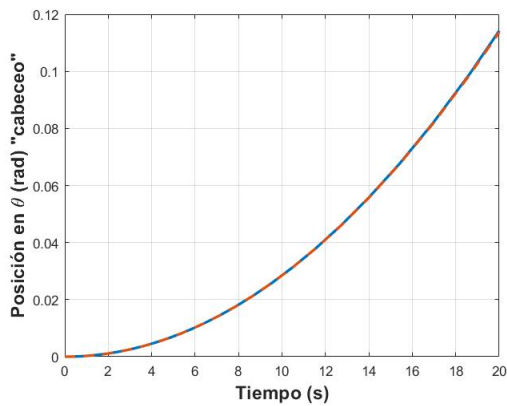
Posición en  $y$  (m).



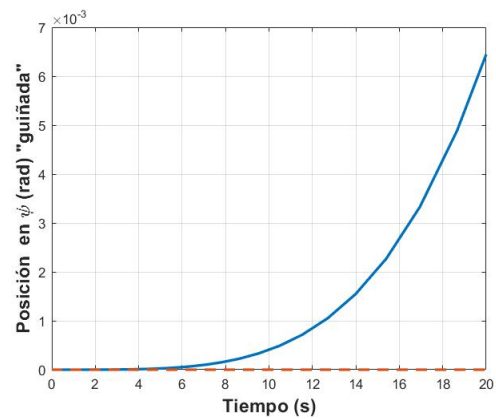
Posición en  $z$  (m).



Posición en  $\phi$  (rad).



Posición en  $\theta$  (rad).



Posición en  $\psi$  (rad).

Figura 2.9: Simulación en lazo abierto para la dinámica completa y linealizada.

## 2.6. Conclusión

De la descripción del modelo matemático es posible notar que: a) un cuadirrotor es un sistema subactuado y para lograr controlar su dinámica es necesario diseñar lazos de control en cascada, b) el modelo matemático obtenido puede presentar tanto no linealidades como indeterminaciones, estas últimas cuando los ángulos  $\theta$  y  $\phi$  son iguales a 90 grados debido al determinante de la matriz  $I_r$ . Por último, en la simulación de la Figura 2.9 se demuestra que la linealización obtenida es fiel a la dinámica completa, siendo solamente  $\psi$  la variable que presenta un comportamiento mínimamente distinto, cabe recordar que las simulaciones son realizadas con los parámetros físicos del AR Drone 2.0.

# CAPÍTULO 3

## ESTRATEGIAS DE CONTROL

# Capítulo 3

## Estrategias de control.

El objetivo de este capítulo es diseñar y simular diferentes estrategias de control para estabilizar la dinámica completa del cuadricóptero AR Drone 2.0. En este capítulo se proponen principalmente algoritmos de control convencionales (sección 3.1) y la técnica de control por modos deslizantes (sección 3.2), los cuales son aplicados al cuadricóptero. Se presentan dos controladores por modos deslizantes, el primero se diseñó partiendo del modelo matemático, mientras que el segundo partiendo de la identificación del control interno del AR Drone 2.0. Finalmente en la sección 3.3 se presentan las conclusiones del capítulo.

### 3.1. Control clásico.

#### 3.1.1 Preliminares.

En la Figura 3.1 se muestra el esquema general de un sistema de control en lazo cerrado, para el cual la señal de error  $e = r - p$  se define como la diferencia existente entre la señal de referencia  $r$  y la señal de salida  $p$  medida a través de los sensores. Como se puede visualizar, la señal de error excita al controlador, este a su vez responde con una manipulación sobre el actuador o bien el elemento final de control y este último realiza un trabajo sobre la planta o proceso. Con esto queda claro que en esta configuración la manipulación es una función del error del sistema.

Pese a que en las últimas décadas se han diseñado técnicas de control avanzadas, con respuestas óptimas y robustas, los controladores proporcional integral y derivativo (PID) siguen siendo los más populares en el sector industrial

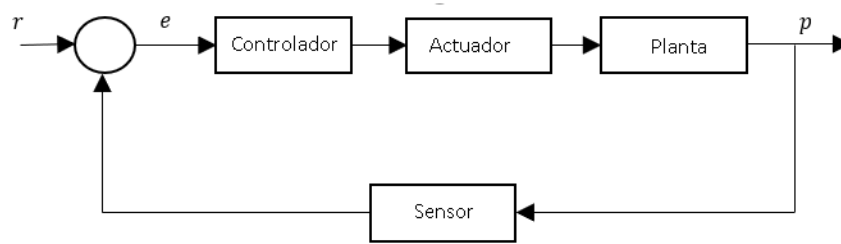


Figura 3.1: Lazo genérico de control.

[41], esto debido a factores como la relación intuitiva que presenta entre sus parámetros y la respuesta del sistema, y también, a que se puede implementar aun cuando no se conoce el modelo matemático de la planta, algo común en campo.

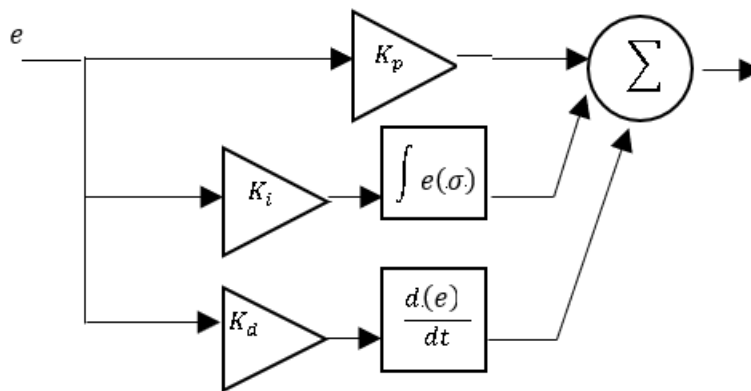


Figura 3.2: Esquema de un PID.

En la Figura 3.2 se muestra el esquema más básico de los controladores PID, en el cual se distinguen las tres acciones de control que lo conforman. La ley de control de un PID está dada por la Ecuación 3.1.

$$u = K_p e + K_i \int_0^t e(\sigma) d\sigma + K_d \frac{de}{dt} = K_p \left( e + \frac{1}{t_i} \int_0^t e(\sigma) d\sigma + t_d \frac{de}{dt} \right). \quad (3.1)$$

donde  $t_i$  y  $t_d$  son los tiempos de integración y derivación respectivamente,  $K_p$ ,  $K_i$ ,  $K_d$  las ganancias proporcional, integral y derivativa, respectivamente y  $e$  el error. A continuación se describen cada una de las acciones de control que lo conforman.

- Proporcional: Esta acción representa una ganancia en trayectoria directa y su modificación tiende a trasladar los polos del lazo de control. En general, el



aumento de  $K_p$  aumentará la velocidad del sistema, pero si es demasiado grande este comenzará a oscilar llegando a inestabilizarlo.

- Proporcional-Derivada: Una manera de evitar las oscilaciones excesivas es agregar la acción derivada, ya que, como la derivada es la pendiente del error, esta nos indica su tendencia y se puede anticipar el comportamiento del sistema dentro de  $t_d$  segundos.
- Proporcional-Integral: En presencia de error en estado estable es recomendable agregar un offset o compensación a la acción de control, de manera tal que compense el error. Una solución automática para resolver este problema es aumentar el valor de la compensación proporcionalmente a la integral del error. La constante  $t_i$  da una idea del tiempo que tardará el controlador en anular el error de estado estable de manera automática.

Ajustar un controlador PID puede parecer fácil, debido a que requiere que se encuentren solamente tres valores: ganancias proporcional  $K_p$ , integral  $K_i$  y derivativa  $K_d$ . De hecho, encontrar de forma segura y sistemática el conjunto de ganancias que garantice el mejor rendimiento de su sistema de control es una tarea compleja. Tradicionalmente, los controladores PID se ajustan manualmente o mediante métodos basados en reglas.

Los métodos manuales son iterativos y se sabe que requieren mucho tiempo, y si se utilizan en el hardware, pueden causar daños. Los métodos basados en reglas también tienen serias limitaciones: no son compatibles con ciertos tipos de modelos de plantas, como plantas inestables, plantas de orden superior o plantas con poco o ningún retraso de tiempo. Además del ajuste, el control PID implica desafíos de diseño e implementación, como la implementación en tiempo discreto y el escalado de punto fijo.

### 3.1.1.1 Índices de desempeño

Para poder validar la calidad del desempeño del controlador, se calculan los índices de desempeño a lo largo de su tiempo de ejecución. El índice Integral del Error Cuadrático (IEC) penaliza los errores de magnitud elevada, el Integral del Valor Absoluto (IVAE) penaliza el error en ambas direcciones, el Integral del

Tiempo del Valor Absoluto (ITVAE) penaliza las variaciones grandes así como la duración de las oscilaciones y por último, el Integral del Tiempo por el Valor Cuadrático (ITEC) penaliza el tiempo durante el cual se mantienen las desviaciones respecto a la referencia.

$$IEC = \int_0^t e(t)^2 dt \quad (3.2)$$

$$IVAE = \int_0^t e(t) dt \quad (3.3)$$

$$ITVAE = \int_0^t te(t)^2 dt \quad (3.4)$$

$$ITEC = \int_0^t te(t) dt \quad (3.5)$$

### 3.1.2 Control clásico para un cuádrirrotor

Debido a que el cuádrirrotor es un sistema sub-actuado, donde la dinámica de rotación es más rápida que la dinámica de translación, se considera para su control un esquema jerárquico. Las variables  $\psi$ ,  $\theta$ ,  $\phi$  y  $z$  pueden controlarse directamente, mientras que,  $x$  y  $y$  se controlan de forma indirecta a través de los ángulos  $\theta$  y  $\phi$  respectivamente.

Se propone un controlador PID considerando un lazo de altitud, un lazo de control interno y un lazo de control externo siguiendo la estructura mostrada en la Figura 3.3, acorde a las Ecuaciones 3.6 - 3.11.

Lazo de altitud:

$$u = k_{p_z} e_z + k_{i_z} \int_0^t e_z(\rho) d\rho - k_{d_z} \dot{z} + mg. \quad (3.6)$$

Control interno:

$$\tau_2 = k_{p_\phi} e_\phi + k_{i_\phi} \int_0^t e_\phi(\rho) d\rho - k_{d_\phi} \dot{\phi}. \quad (3.7)$$

$$\tau_3 = k_{p_\theta} e_\theta + k_{i_\theta} \int_0^t e_\theta(\rho) d\rho - k_{d_\theta} \dot{\theta}. \quad (3.8)$$

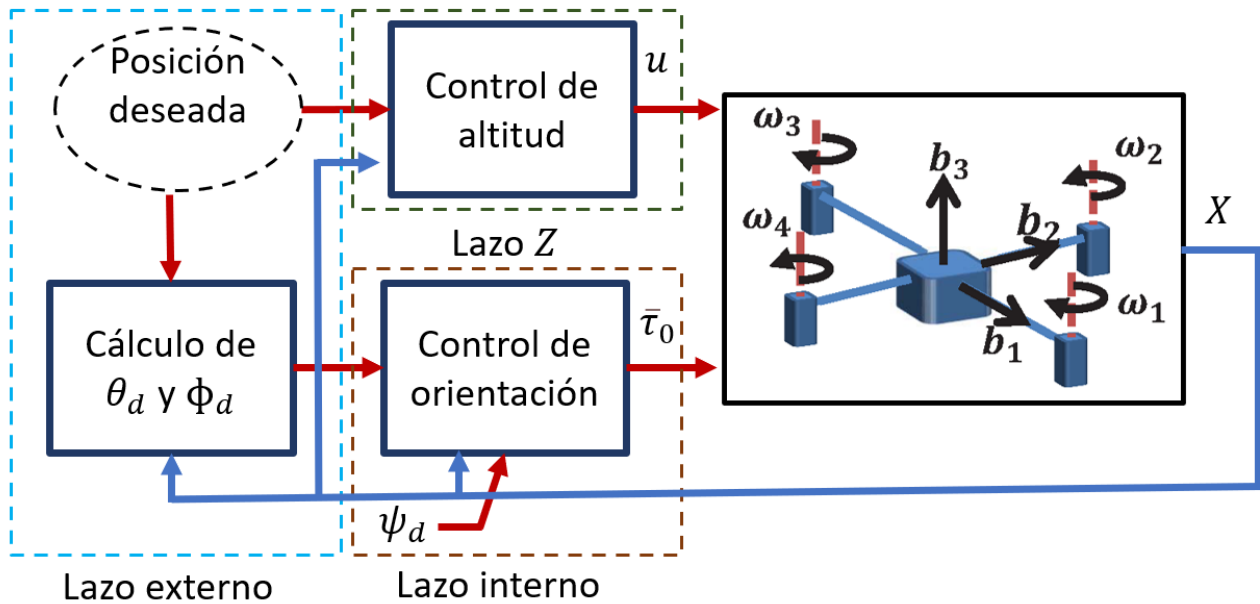
$$\tau_4 = k_{p_\psi} e_\psi + k_{i_\psi} \int_0^t e_\psi(\rho) d\rho - k_{d_\psi} \dot{\psi}. \quad (3.9)$$

Control externo:

$$\phi_d = k_{px}e_x + k_{ix} \int_0^t e_x(\rho) d\rho - k_{dx}\dot{x}. \quad (3.10)$$

$$\theta_d = -(k_{py}e_y + k_{iy} \int_0^t e_y(\rho) d\rho - k_{dy}\dot{y}). \quad (3.11)$$

donde  $e_x = x - x_d$ ,  $e_y = y - y_d$ ,  $e_z = z - z_d$ ,  $e_\phi = \phi - \phi_d$ ,  $e_\theta = \theta - \theta_d$  y  $e_\psi = \psi - \psi_d$ .



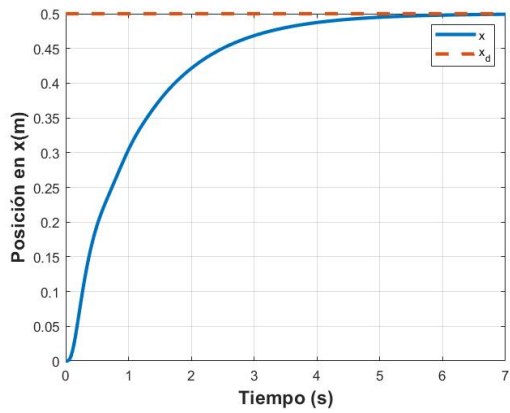
$$\bar{\tau}_0 = [\tau_2, \tau_3, \tau_4]^T \quad X = [x, \dot{x}, y, \dot{y}, z, \dot{z}, \theta, \dot{\theta}, \phi, \dot{\phi}, \psi, \dot{\psi}]^T$$

Figura 3.3: Lazo de control para dinámica linealizada.

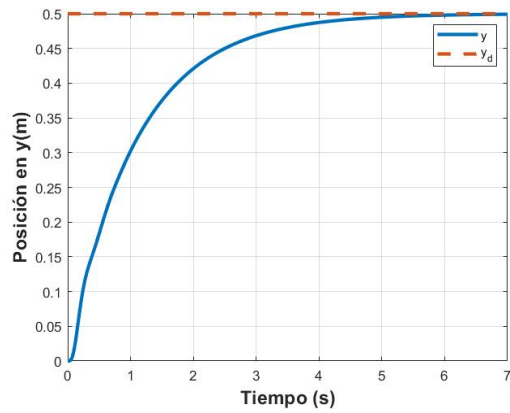
### Regulación

En la Figura 3.4 se muestran los resultados numéricos de la técnica de control PID aplicada al modelo lineal del cuadricóptero. Se observa que las posiciones  $x$  y  $y$  presentan una respuesta suave y se estabilizan rápidamente en menos de 5 segundos. La posición en  $z$  es más rápida y casi en 1 segundo ya se estabilizó, presentando también una respuesta suave. Respecto a la dinámica rotacional, los ángulos de Euler se estabilizarán rápidamente a cero.

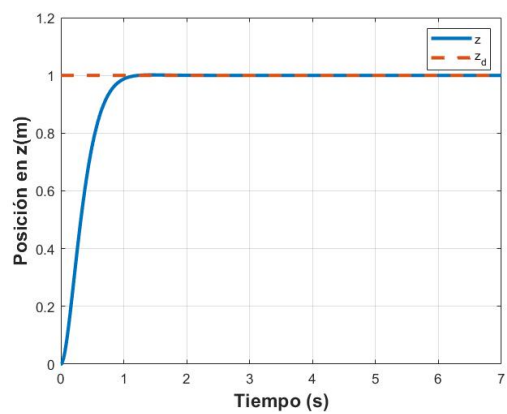
En la Figura 3.5 se aprecian las señales de control de la técnica de control PID aplicada al modelo lineal del cuadricóptero, se visualiza que la fuerza de empuje total  $u$  se estabiliza en  $mg$  rápidamente y que los torques llegan a cero en menos de 1 segundo.



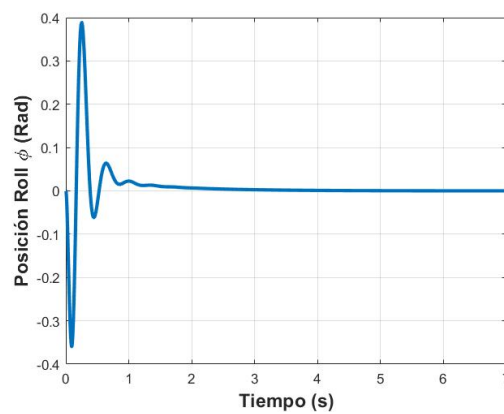
Posición en  $x$  (m).



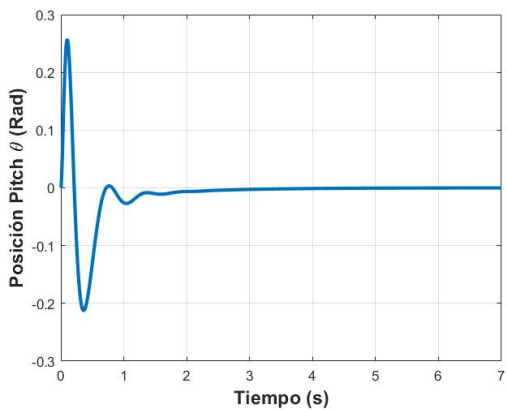
Posición en  $y$  (m).



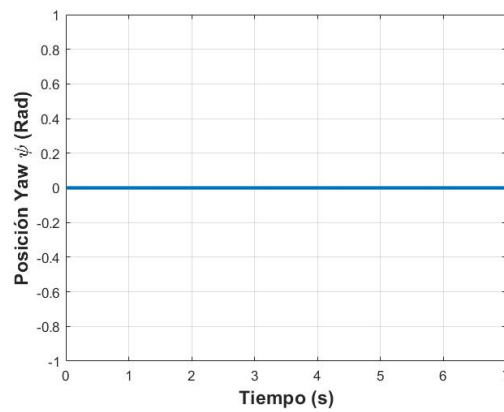
Posición en  $z$  (m).



Posición en  $\phi$  (rad).



Posición en  $\theta$  (rad).



Posición en  $\psi$  (rad).

Figura 3.4: Vector de estado del PID en dinámica linealizada.

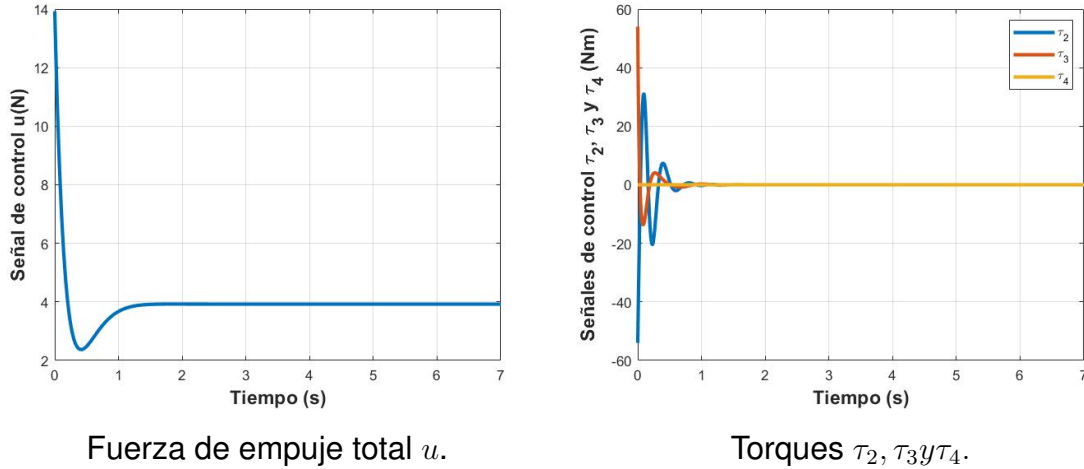


Figura 3.5: Señales de control del PID en dinámica linealizada.

Además, se propone también un controlador PID para la dinámica completa, con la estructura mostrada en la Figura 3.6, dado  $e_X = X_d - X$ ,  $X = [x, y, z, \phi, \theta, \psi]^T$ ,  $X_d = [x_d, y_d, z_d, \phi_d, \theta_d, \psi_d]^T$ , debido a que se propone un control suave y que las referencias de posición se suponen constantes  $\dot{e}_X \cong -\dot{X}$ . La señal de control se puede representar como:

$$u_* = K_p(X_d - X) - K_v\dot{X} + K_i \int_0^t e_X(\rho) d\rho + [0, 0, mg, 0, 0, 0]^T. \quad (3.12)$$

donde  $u_* = [u_1, u_2, u, \tau_2, \tau_3, \tau_4]^T$  contiene a las señales virtuales  $u_x$  y  $u_y$ ;  $x_d, y_d$  y  $z_d$  son valores constantes de referencia determinados por el usuario,  $\psi_d$  se elige cero,  $\phi_d$  y  $\theta_d$  se calculan por:

$$\phi_d = \sin^{-1} \frac{u_1 - \sin(\phi) \sin(\psi_d)}{\cos(\psi_d) \cos(\phi)} \quad (3.13)$$

$$\theta_d = \sin^{-1}(\sin(\phi)u_1 - \cos(\phi)u_2) \quad (3.14)$$

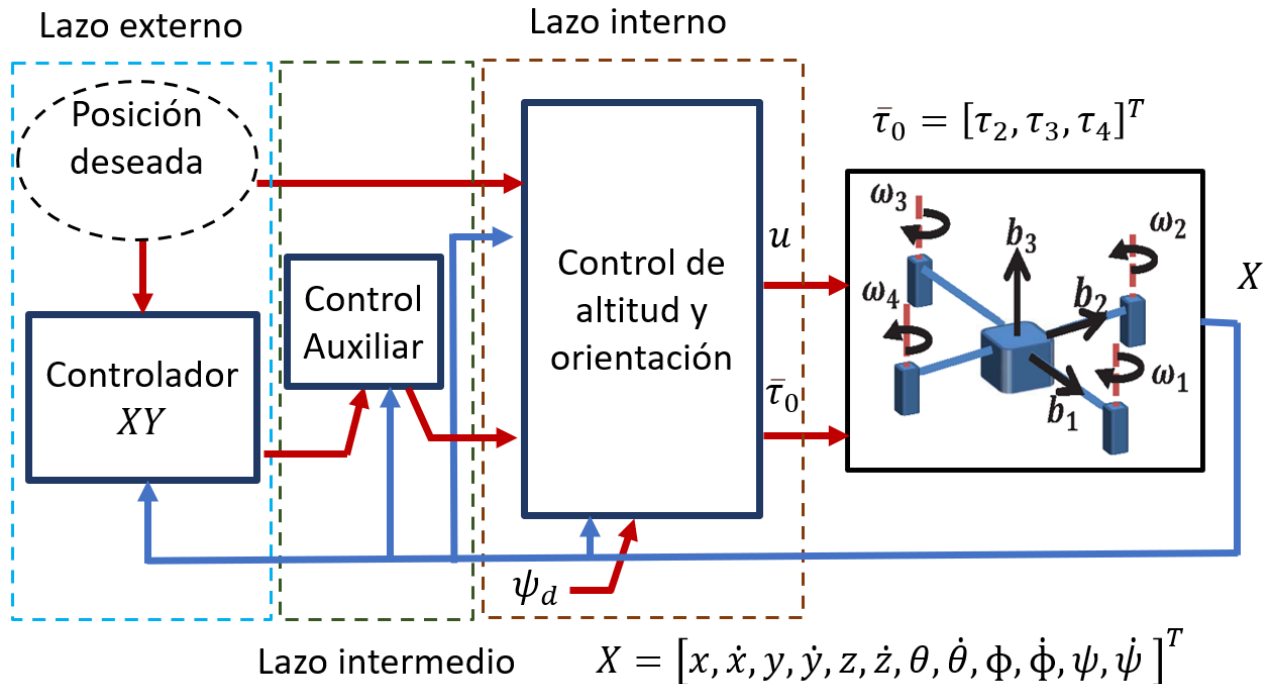


Figura 3.6: Estructura del lazo de control.

En la Figura 3.7 se presentan los resultados obtenidos de la técnica de control PID aplicada al modelo no lineal del cuadricóptero. Se puede visualizar que las posiciones  $x$  y  $y$  presentan una respuesta suave y se estabilizan rápidamente en menos de 2 segundos. La posición en  $z$  es más rápida y casi en 1 segundo ya se estabilizó, presentando también una respuesta suave. Respecto a la dinámica rotacional, los ángulos de Euler se estabilizan rápidamente a cero. Comparando los resultados del control PID en el modelo lineal y en el modelo no lineal, se observa que el control en el modelo no lineal tiene en general un mejor desempeño debido a que las posiciones traslacionales se estabilizan en un menor tiempo y las magnitudes de los ángulos de rotación son menores.

En la Figura 3.8 se aprecian las señales de control de la técnica de control PID aplicada al modelo no lineal del vehículo, se visualiza que la fuerza de empuje total  $u$  se estabiliza en  $mg$  rápidamente y que los torques llegan a cero en menos de 1 segundo.

En las Tablas 3.1 y 3.2 se muestran los índices de desempeño obtenidos para ambos controladores. Para las sintonizaciones presentadas se obtuvo un mejor desempeño en el controlador PID aplicado a la dinámica completa. La principal

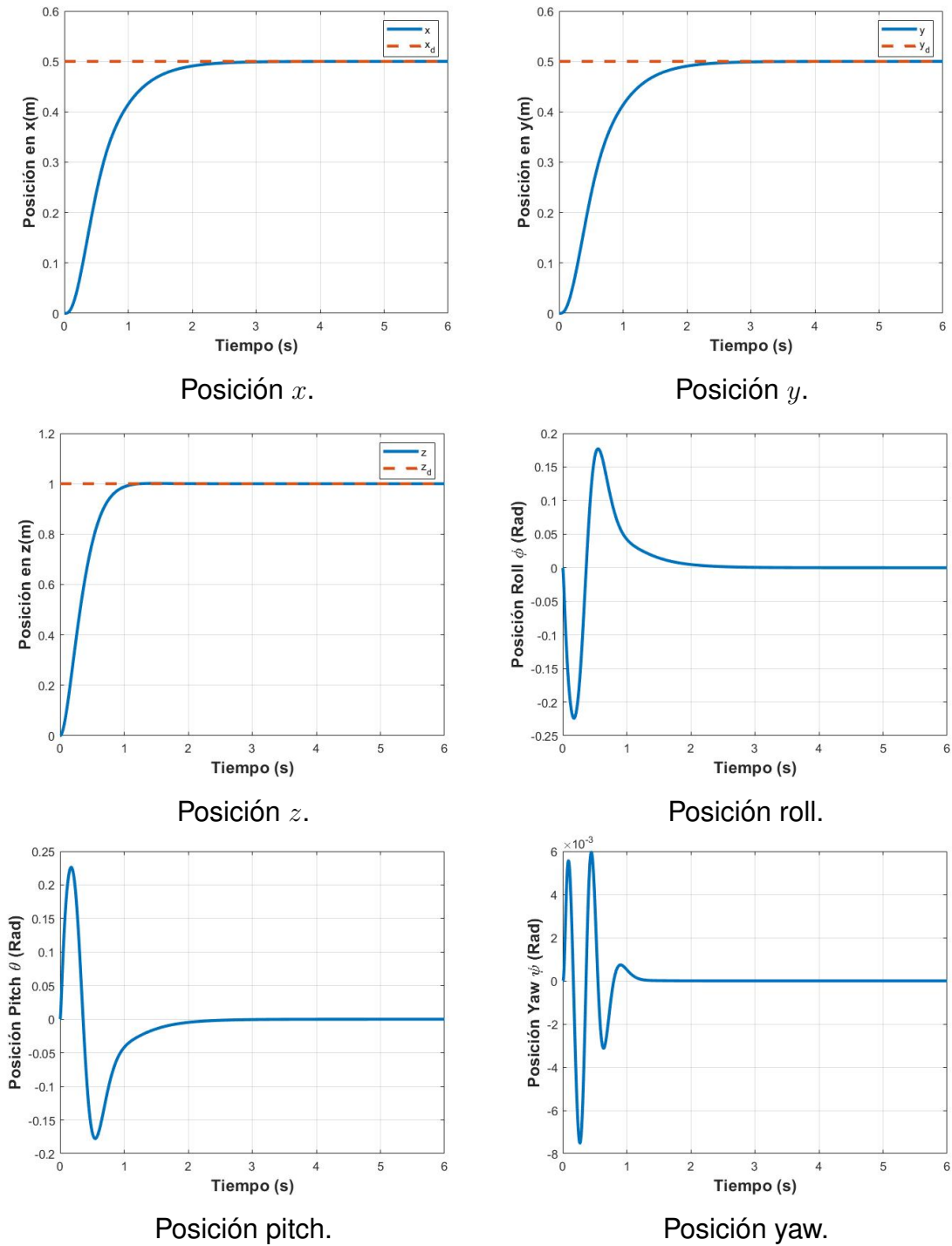


Figura 3.7: Vector de estado del PID en dinámica completa.

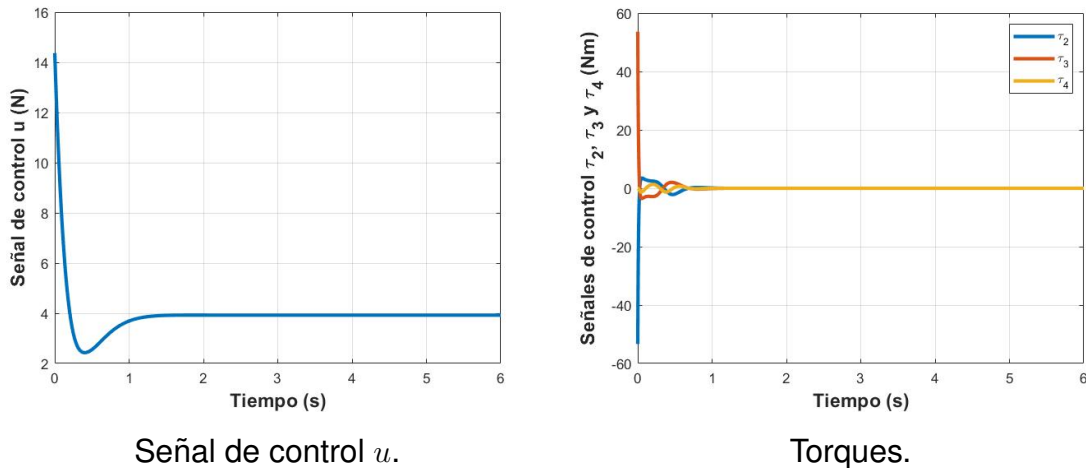


Figura 3.8: Señales de control del PID en dinámica completa.

diferencia en los controladores se encuentra en los desplazamientos angulares, ya que es sobre ellos que se aplica la linealización.

Tabla 3.1: Índices de desempeño en PID modelo linealizado.

Índice	$x$	$y$	$z$	$\phi$	$\theta$	$\psi$
IEC	0.1386	0.1385	0.2356	0.0400	0.0303	0.0000
IVAE	0.5432	0.5451	0.3618	0.1201	0.1152	0.0000
ITVAE	0.5848	0.5879	0.0933	0.0261	0.0328	0.0000
ITEC	0.0724	0.0735	0.0386	0.0052	0.0040	0.0000

Tabla 3.2: Índices de desempeño en PID modelo completo.

Índice	$x$	$y$	$z$	$\phi$	$\theta$	$\psi$
IEC	0.1019	0.1029	0.2321	0.0046	0.0047	0.0000
IVAE	0.3208	0.3235	0.3577	0.0536	0.0541	0.0028
ITVAE	0.1556	0.1579	0.0907	0.0209	0.0208	0.0011
ITEC	0.0282	0.0287	0.0376	0.0011	0.0011	0.0000

### Seguimiento de trayectoria

Se realizó una segunda simulación con el propósito particular de validar el seguimiento de trayectoria del controlador PID con el modelo de la dinámica



completa, para lo cual se consideraron las siguientes referencias:  $x_d = r \cos(\gamma) - r$ ,  $y_d = r \sin(\gamma)$  y  $z_d = r \sin(\gamma) + 1$ , con  $r = 0.5$ , los resultados se muestran en la Figura 3.9.

En Figura 3.9 se puede observar que el propósito de control se logró debido a que se tiene un rápido y suave seguimiento de trayectoria en las posiciones translacionales (en azul se presenta el estado y en naranja su referencia).

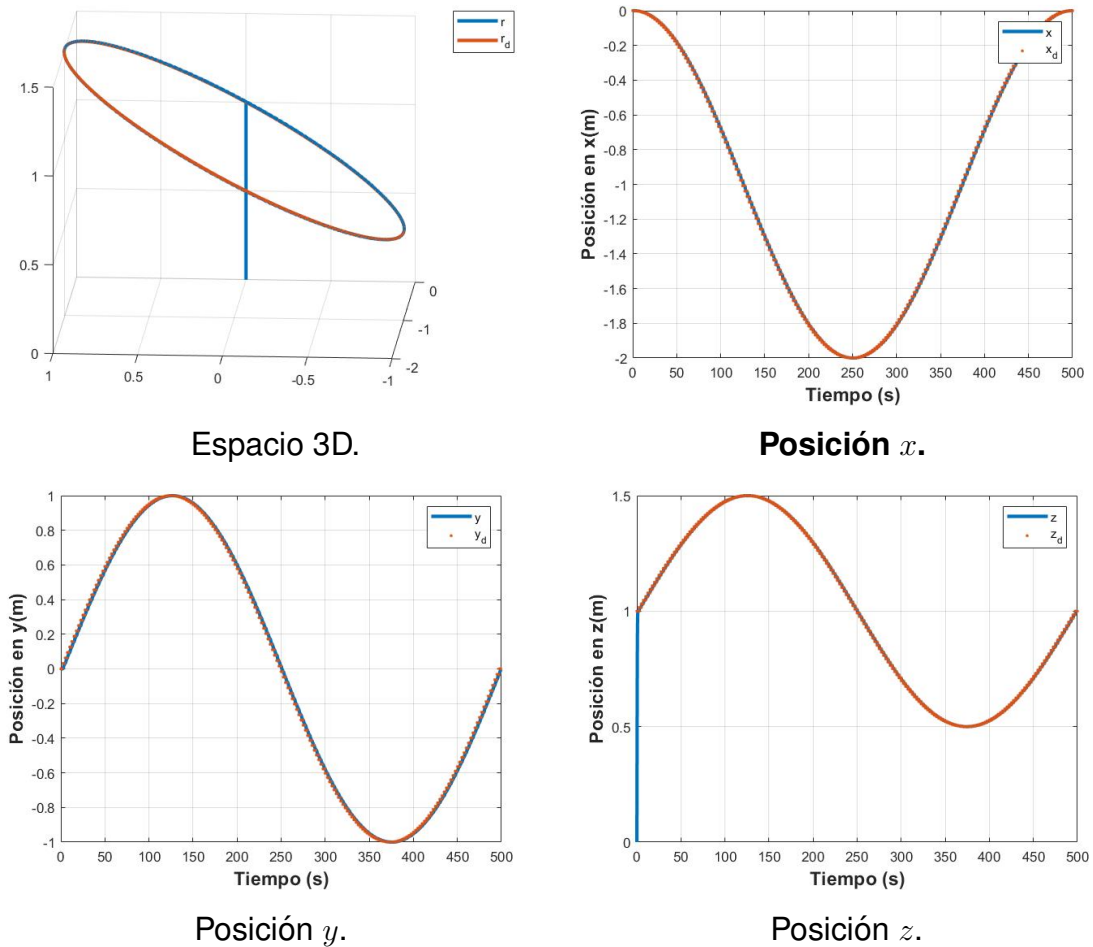


Figura 3.9: Simulación PID con seguimiento de trayectoria.

## 3.2. Control por modos deslizantes.

### 3.2.1 Preliminares.

El término modo deslizante apareció primero en el contexto de los sistemas relevadores, apareciendo en el control como una función del switch o interruptor de estado de sistema en alta frecuencia (teóricamente); este movimiento se llama modo deslizante. La idea del control por modos deslizantes (CMD) se basa en la introducción de una función personalizada, denominada variable deslizante. Al tiempo que la variable de deslizamiento se vuelve igual a cero, se define la superficie deslizante.

El diseño adecuado de la variable deslizante produce un rendimiento adecuado del sistema en lazo cerrado, mientras que las trayectorias del sistema pertenecen dentro de la superficie deslizante. La idea del CMD es dirigir la trayectoria del sistema a la superficie deslizante y posteriormente mantener el movimiento de la superficie a partir de entonces, por medio del control. Las características principales del modo deslizante son su insensibilidad a perturbaciones externas e internas emparejadas a la señal de control, la máxima precisión y la convergencia en tiempo finito de las variables deslizantes a cero [43].

El primer paso para el diseño de un CMD es la selección de una función de conmutación para el movimiento del sistema dentro de la superficie deslizante, y posterior a esto se diseña una ley de control que haga que la superficie deslizante sea atractiva para el sistema en presencia de perturbaciones e incertidumbres. Para lograr su régimen de funcionamiento debido al uso de funciones signo (conmutación) en su diseño, este controlador añade un efecto no deseado: el castaño, que consiste en oscilaciones de alta frecuencia. Este problema se trata de resolver ya sea *a)* aproximando la función signo de naturaleza discontinua a una función algebraica continua que dependa de la superficie deslizante, *b)* diseñando controladores CMD de orden superior.

En la estrategia CMD se hace uso de la función de conmutación signo  $sgn(s)$ , la cual se define como:

$$sgn(s) \approx \frac{s}{s + \zeta}, \zeta \rightarrow 0 \quad (3.15)$$

Del modelo matemático en  $\{I\}$ :

$$m\ddot{r} = \begin{bmatrix} 0 \\ 0 \\ -mg \end{bmatrix} + \begin{bmatrix} c_\psi s_\theta + c_\theta s_\phi s_\psi \\ s_\psi s_\theta - c_\theta s_\phi c_\psi \\ c_\theta c_\phi \end{bmatrix} u, \quad (3.16)$$

$$I_r \ddot{\eta} = \bar{\tau}_0 + \bar{C} \quad (3.17)$$

donde

$$\bar{C} = [C_1, C_2, C_3]^T, \quad (3.18)$$

$$I_r = \begin{pmatrix} I_{xx} c_\theta & 0 & -I_{xx} c_\phi s_\theta \\ 0 & I_{yy} & I_{yy} s_\phi \\ I_{zz} s_\theta & 0 & I_{zz} c_\theta c_\phi \end{pmatrix}, \quad (3.19)$$

con

$$C_1 = [I_{xx} + I_{yy} - I_{zz}] \dot{\phi} \dot{\theta} s_\theta + [-I_{xx} + I_{yy} - I_{zz}] \dot{\phi} \dot{\psi} s_\theta s_\phi + [I_{xx} + I_{yy} - I_{zz}] \dot{\phi} \dot{\psi} c_\theta c_\phi + [I_{yy} - I_{zz}] \dot{\psi}^2 s_\phi c_\phi c_\theta,$$

$$C_2 = [-I_{yy} + [-I_{xx} + I_{zz}] c_{2\theta}] \dot{\phi} \dot{\psi} c_\phi + [-I_{xx} + I_{zz}] [\dot{\phi}^2 - \dot{\psi}^2 c_\phi^2] c_\theta s_\theta \mathbf{y}$$

$$C_3 = [I_{xx} - I_{yy} - I_{zz}] \dot{\phi} \dot{\theta} c_\theta + [I_{xx} - I_{yy} + I_{zz}] \dot{\phi} \dot{\psi} s_\phi c_\theta + [-I_{xx} + I_{yy} + I_{zz}] \dot{\phi} \dot{\psi} s_\theta c_\phi + [-I_{xx} + I_{yy}] \dot{\psi}^2 s_\phi c_\phi s_\theta.$$

Una representación alternativa de las Ecuaciones 3.16 y 3.17 de donde parte el control por modos deslizantes es:

$$\ddot{X} = f + u_T = v, \quad (3.20)$$

donde:

$$\ddot{X} = \begin{bmatrix} \ddot{x} & \ddot{y} & \ddot{z} & \ddot{\phi} & \ddot{\theta} & \ddot{\psi} \end{bmatrix}^T,$$

$$v = \begin{bmatrix} v_x & v_y & v_z & v_\phi & v_\theta & v_\psi \end{bmatrix}^T,$$

$$f = \begin{bmatrix} mI_{3 \times 3} & 0_{3 \times 3} \\ 0_{3 \times 3} & I_r \end{bmatrix}^{-1} \begin{bmatrix} 0 \\ 0 \\ 0 \\ -mg \\ C_1 \\ C_2 \\ C_3 \end{bmatrix}, \quad (3.21)$$

$$u_T = \begin{bmatrix} mI_{3 \times 3} & 0_{3 \times 3} \\ 0_{3 \times 3} & I_r \end{bmatrix}^{-1} \begin{bmatrix} (c_\psi s_\theta + c_\theta s_\phi s_\psi) u \\ (s_\psi s_\theta - c_\theta s_\phi c_\psi) u \\ (c_\theta c_\phi) u \\ \tau_2 \\ \tau_3 \\ \tau_4 \end{bmatrix}, \quad (3.22)$$

donde  $0_{3 \times 3}$  es una matriz de ceros y  $I_{3 \times 3}$  es la matriz identidad. En la Ecuación (3.20)  $v \in \mathbb{R}^6$  se utiliza como entrada auxiliar de control.

Sea el error del sistema  $\tilde{x} = X_d - X$ , con  $X = [x, y, z, \phi, \theta, \psi]^T$  y  $X_d = [x_d, y_d, z_d, \phi_d, \theta_d, \psi_d]^T$ . Se define la superficie deslizando como

$$s_i = \dot{\tilde{x}}_i + \lambda \tilde{x}_i + \alpha_i \int \tilde{x}_i, \quad (3.23)$$

donde  $\tilde{x}_i$  representa al  $i$ -ésimo elemento en el vector de error  $\tilde{x}$

El objetivo del algoritmo de control es lograr que en un tiempo  $t_r$ , mediante la acción de  $v$ , el sistema evolucione hacia la superficie de deslizamiento, y que permanezca allí después de éste.

La derivada temporal se  $s$  se obtiene de la siguiente manera:

$$\frac{d}{dt}(s) = \frac{d}{dt} \left( \dot{\tilde{x}} + K_\lambda \tilde{x} + K_\alpha \int \tilde{x} \right), \quad (3.24)$$

$$\dot{s} = \ddot{\tilde{x}} + K_\lambda \dot{\tilde{x}} + K_\alpha \tilde{x}, \quad (3.25)$$

$$\dot{s} = \ddot{x}_d - \ddot{x} + K_\lambda \dot{\tilde{x}} + K_\alpha \tilde{x}, \quad (3.26)$$

$$\dot{s} = \ddot{x}_d - v + K_\lambda \dot{\tilde{x}} + K_\alpha \tilde{x}, \quad (3.27)$$

Donde  $K_\lambda$  y  $K_\alpha$  son matrices diagonales y sus elementos en la diagonal son  $\lambda_i$  y  $\alpha_i$ .

### 3.2.2 Control por modos deslizando aplicado a un cuadrirrotor

También, se desarrolló y simuló con los mismos parámetros del móvil mostrados en la Tabla 2.1, un controlador por modos deslizando, con base a la estructura de control de la Figura 3.6 y fundamentado por las Ecuaciones (3.28) - (3.35).

Lazo intermedio:

$$v = \bar{u} + Ks + Q\text{sing}(s), \quad (3.28)$$

donde

$$\bar{u} = \ddot{x}_d + K_\lambda \dot{\tilde{x}} + K_\alpha \tilde{x} \quad (3.29)$$

Lazo interno:

$$u = m\sqrt{v_x^2 + v_y^2 + [v_z + g]^2}, \quad (3.30)$$

$$\begin{pmatrix} \tau_2 \\ \tau_3 \\ \tau_3 \end{pmatrix} = \begin{pmatrix} v_\phi \\ v_\theta \\ v_\psi \end{pmatrix} - \begin{pmatrix} K_1 \\ K_2 \\ K_3 \end{pmatrix}, \quad (3.31)$$

Lazo externo

$$\phi_d = \tan^{-1} \left[ \frac{v_x s_{\psi_d} - v_y c_{\psi_d}}{v_z + g} \right], \quad (3.32)$$

$$\theta_d = \tan^{-1} \left[ \frac{v_x c_{\psi_d} + v_y s_{\psi_d}}{\sqrt{(v_x s_{\psi_d} - v_y c_{\psi_d})^2 + (v_z + g)^2}} \right]. \quad (3.33)$$

con

$$\text{sing}(s) \approx \frac{s}{s + \zeta}, \zeta \rightarrow 0 \quad (3.34)$$

$$s_i = \dot{\tilde{x}}_i + \lambda \tilde{x}_i + \alpha_i \int \tilde{x}_i \quad (3.35)$$

donde  $\lambda$  y  $\alpha_i$  son constantes.

### Regulación

Los resultados obtenidos de la simulación numérica del control por modos deslizantes aplicado al cuadrirrotor y considerando solamente control por regulación se muestran en la Figuras 3.10 y 3.11. En la Figura 3.10 se aprecia que las posiciones  $x$  y  $y$  también poseen una respuesta suave y rápida, debido a que se estabilizan en menos de 4 segundos. Respecto a la posición en  $z$  se estabiliza en menos de 3 segundos y también presenta una respuesta suave. En lo que se refiere a la dinámica rotacional, los ángulos de Euler se estabilizan rápidamente a cero y presentan menores magnitudes que las obtenidas en el control convencional.

En la Figura 3.11 se aprecian las señales de control de la técnica de control CMD aplicada al modelo del vehículo, se visualiza que la fuerza de empuje total  $u$  se estabiliza en  $mg$  rápidamente y que los torques llegan a cero en menos de 2.5 segundos.

Cabe mencionar que con estas respuestas aún no se puede apreciar ninguna ventaja del control por modo deslizante ante el control convencional debido a que no se están considerando perturbaciones.

En la Tabla 3.3 se pueden observar los índices de desempeño de esta estrategia de control.

Tabla 3.3: Índices de desempeño en CMD.

Índice	$x$	$y$	$z$	$\phi$	$\theta$	$\psi$
IEC	0.1750	0.1753	0.4214	0.0004	0.0004	0.0000
IVAE	0.5442	0.5441	0.7187	0.0163	0.0166	0.0001
ITVAE	0.4496	0.4488	0.4374	0.0091	0.0097	0.0000
ITEC	0.0819	0.0821	0.1350	0.0001	0.0001	0.0000

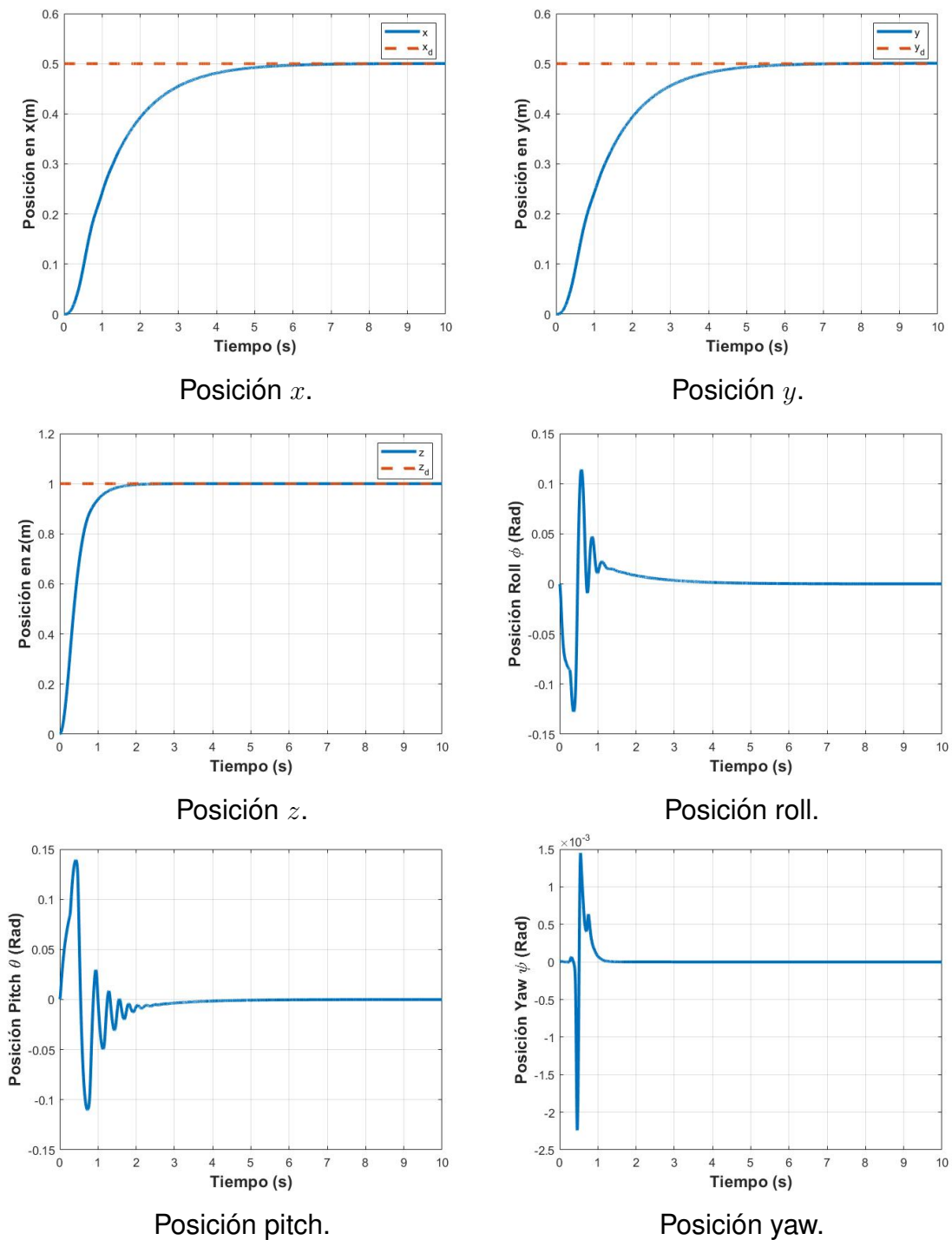


Figura 3.10: Vector de estado del CMD.

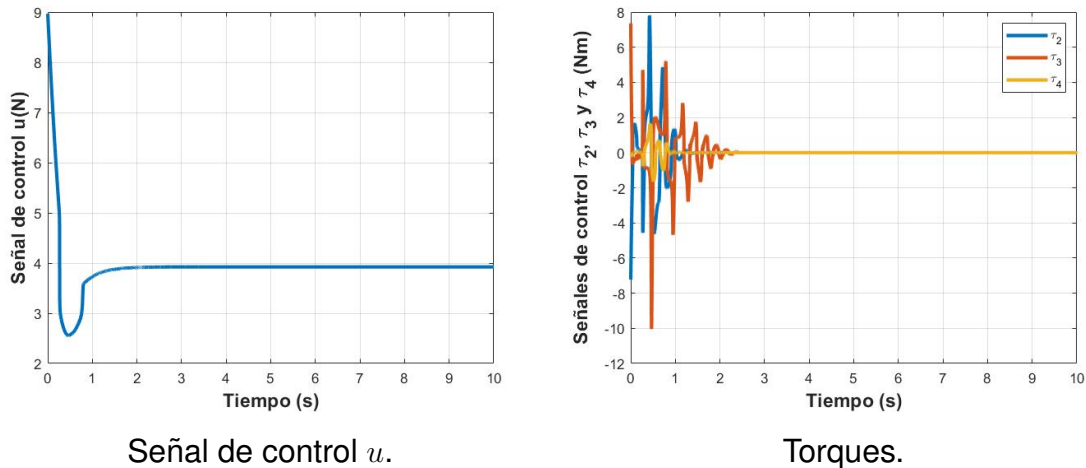


Figura 3.11: Señales de control del CMD.

### Seguimiento de trayectoria

De manera similar que con el control PID, se validó el seguimiento de trayectoria del controlador CMD con la dinámica completa, para la cual se consideraron las mismas referencias que en el control PID, es decir,  $x_d = r \cos(\gamma) - r$ ,  $y_d = r \sin(\gamma)$  y  $z_d = r \sin(\gamma) + 1$ , con  $r = 0.5$ . Los resultados de esta simulación se muestran en la Figura 3.12.

En la Figura 3.12 se puede observar que el propósito de control se logró debido a que se tiene un rápido y suave seguimiento de trayectoria en las posiciones translacionales, en la gráfica tridimensional se puede apreciar el buen seguimiento de trayectoria por parte del vehículo aéreo.



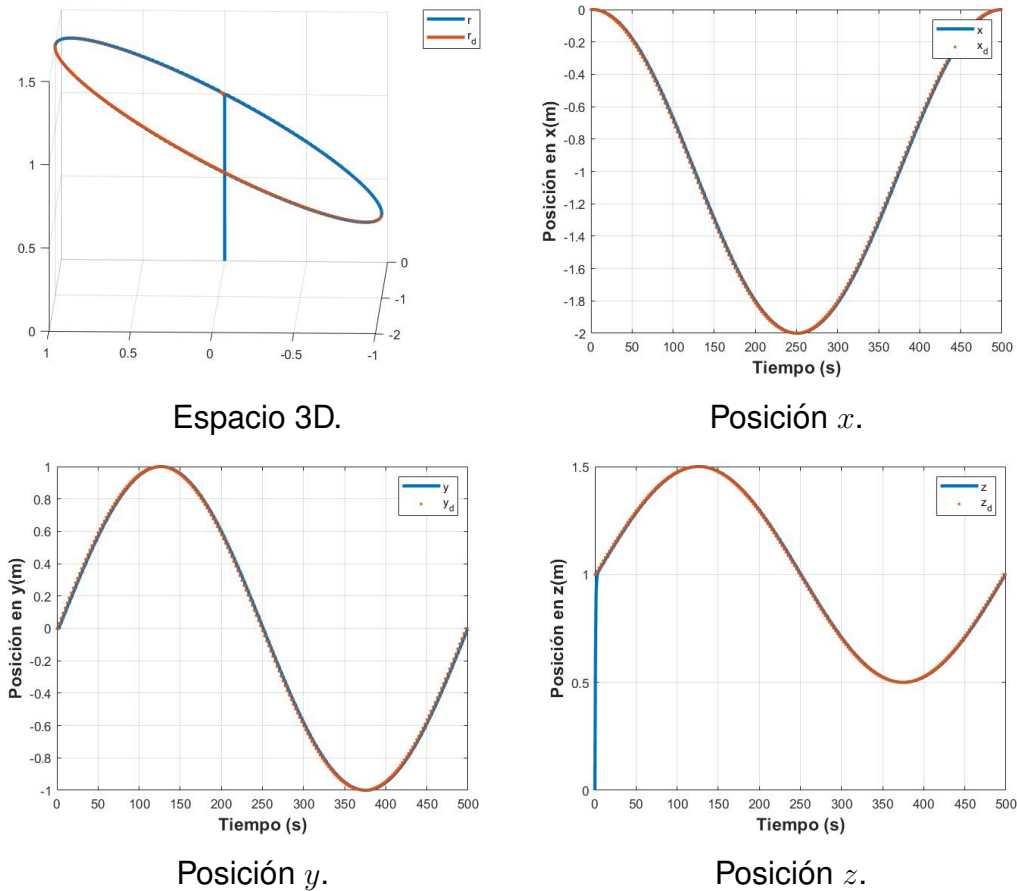


Figura 3.12: Simulación CMD con seguimiento de trayectoria.

### 3.2.3 Control por modos deslizantes en Parrot AR Drone.

En esta subsección se diseña un control por modos deslizantes al vehículo AR Drone de la marca parrot. El AR Drone 2.0 es un cuadricóptero comercializado de uso recreativo por la empresa Parrot, en este cuadricóptero el semi eje positivo  $x$  se encuentra dirigido entre el motor 1 y 2, y el semi eje positivo  $y$  entre el rotor 1 y 4 [33]. En el año 2012 la empresa publicó un kit de desarrollo base Linux (SDK, por sus siglas en inglés), mediante el cual se diseñan aplicaciones que controlan al móvil en tiempo real comunicándose mediante Wi-Fi.

El SDK cuenta con lazos de control internos, de tal manera se puede considerar al AR Drone 2.0 como una caja negra, que dadas las entradas de control  $u_d = [\phi_d, \theta_d, \psi_d, z_d]^T$  acotadas en  $[-1, 1]$ , genera la respuesta  $X = [\dot{x}, \dot{y}, \dot{z}, z, \phi, \theta, \psi]^T$ , como

se visualiza en la Figura 3.13.

Se sabe que la dinámica del control angular del AR Drone es de segundo orden y la relación entre la rotación y la traslación en las parejas  $(\phi, y)$  y  $(\theta, x)$  es de primer orden según lo establecido en [45].



Figura 3.13: Conceptualización del AR Drone 2.0.

En [33] y [34] se calculó el tiempo de retardo en la transmisión de la información en 0.26 segundos y se obtienen aproximaciones por mínimos cuadrados de la función de transferencia que describen estas relaciones, auxiliado del sistema de localización OptiTrack, . Sus resultados se presentan a continuación:

$$\frac{z}{u_z} = \frac{3.384}{s^2 + 3.613s} \quad (3.36)$$

$$\frac{\psi}{u_\psi} = \frac{3.828}{s^2 + 4.225s} \quad (3.37)$$

$$\frac{\theta}{\theta_d} = \frac{87.46}{s^2 + 5s + 33.94} \quad (3.38)$$

$$\frac{v_x}{\theta} = \frac{-0.2168}{s + 0.9906} \quad (3.39)$$

$$\frac{\phi}{\phi_d} = \frac{61.64}{s^2 + 3.2s + 23.31} \quad (3.40)$$

$$\frac{v_y}{\phi} = \frac{0.1991}{s + 0.5469} \quad (3.41)$$

Para las dinámicas representadas por las Ecuaciones (3.36) y (3.37) se propone un controlador proporcional, mientras que para las representadas de (3.38) - (3.41) se propone un controlador cuasi CMD y uno cuasi CMD asintótico [43]. Para el cuasi

CMD corresponde el siguiente vector de estado:

$$\begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{pmatrix} = \begin{pmatrix} x \\ \dot{x}_1 \\ \theta \\ \dot{x}_3 \end{pmatrix}$$

Entonces:

$$\dot{x}_4 = -33.94x_3 - 5x_4 + 87.46u_x + f; \quad (3.42)$$

$$u_x = -c_x(x_2) - p_x \text{sing}(x_2 + c_x(x_{1d} - x_1)); \quad (3.43)$$

$$u_x = \text{sat}(u_x); \quad (3.44)$$

Considerando:

$$\begin{pmatrix} x_5 \\ x_6 \\ x_7 \\ x_8 \end{pmatrix} = \begin{pmatrix} y \\ \dot{x}_5 \\ \phi \\ \dot{x}_7 \end{pmatrix}$$

$$u_y = -c_y(x_6) - p_y \text{sing}(x_6 + c_y(x_{5d} - x_5)); \quad (3.45)$$

$$u_x = \text{sat}(u_x); \quad (3.46)$$

con la función signo aproximada a:

$$\text{sing}(g) \approx \frac{g}{g + \zeta}, \zeta \rightarrow 0 \quad (3.47)$$

Las posiciones translacionales y la posición angular  $\psi$  se muestran en la Figura 3.14, donde se observan respuestas rápidas, sin embargo la posición en  $y$  muy leves oscilaciones alrededor de la referencia deseada. Los índices de desempeño se muestran en la Tabla 3.4. Cabe recordar que la señal de control no es propiamente el torque, sino el ángulo deseado indicado de manera porcentual, donde 1 corresponde 100 por ciento del ángulo máximo configurado y -1 corresponde al cien por ciento del ángulo mínimo.

En la Figura 3.15 se muestran las señales de control, en ambas, es posible ver el efecto castaño generado por este tipo de controlador.

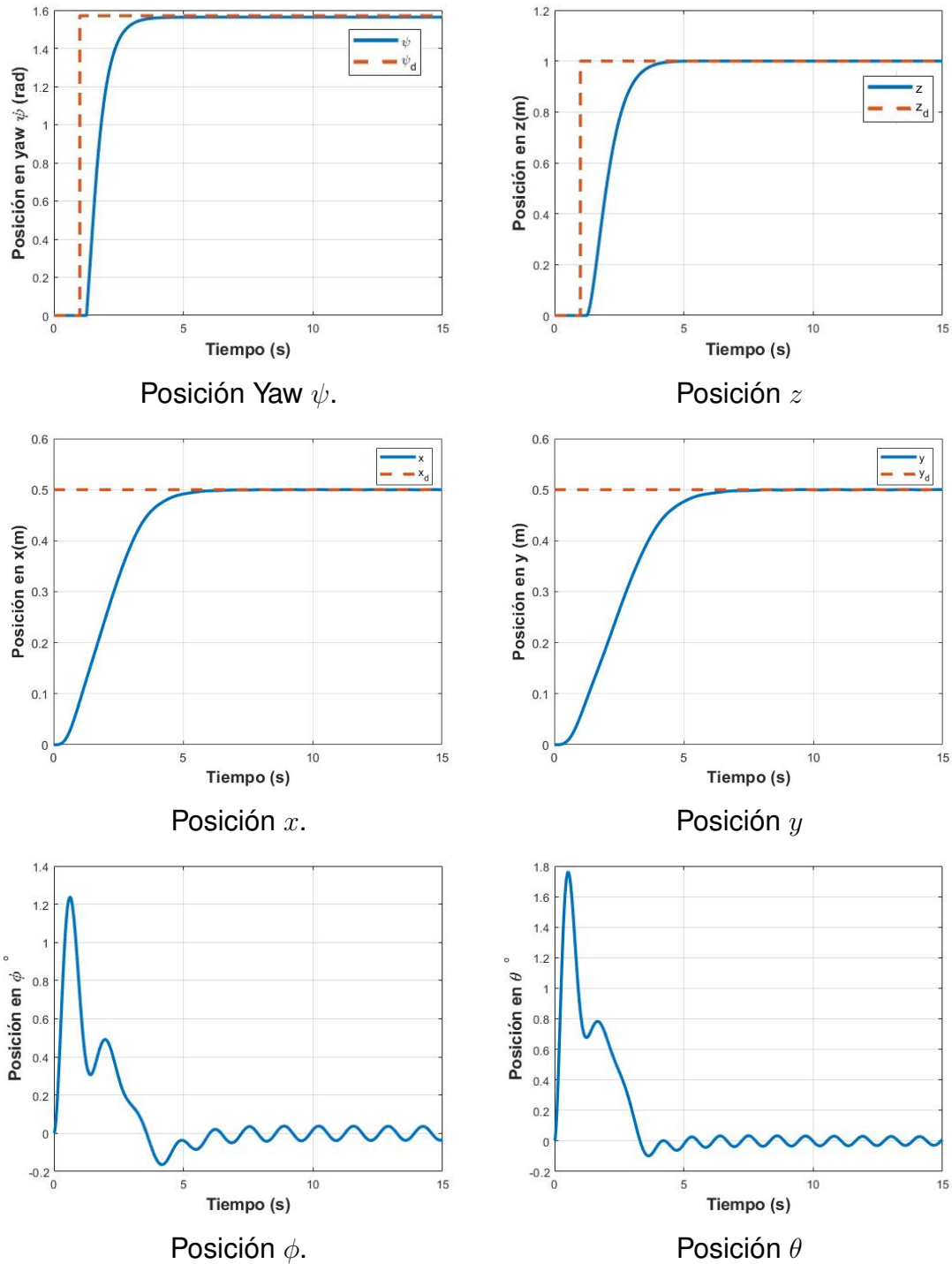


Figura 3.14: Posiciones del Cuasi-CMD y control P.

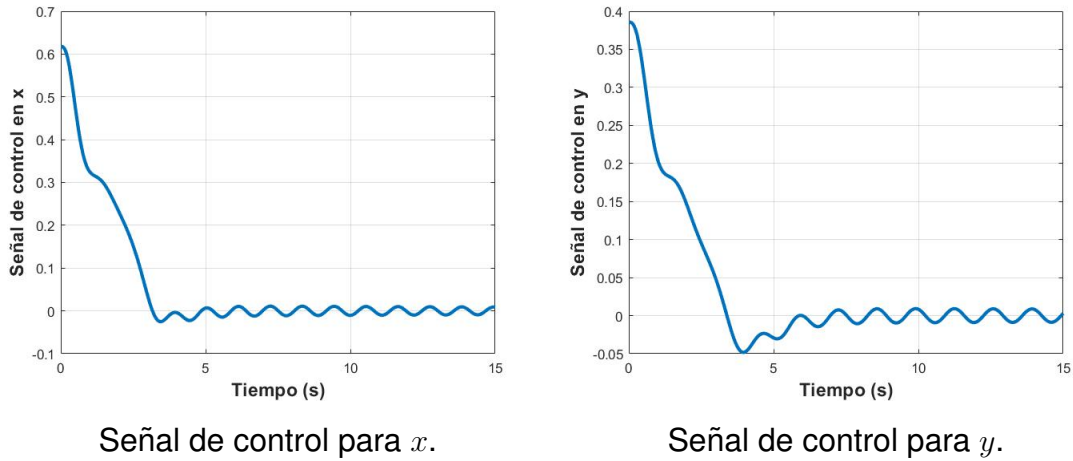


Figura 3.15: Señales de control del Cuasi-CMD.

Tabla 3.4: Índices de desempeño en cuasi CMD.

Índice	$x$	$y$
IEC	0.3781	0.4545
IVAE	1.0728	1.2797
ITVAE	1.4823	2.0902
ITEC	0.3619	0.5187

En segunda instancia el controlador cuasi CMD asintótico, con el fin de reducir el castaño, el control se ve afectado por una perturbación  $f = A \sin(\omega_f t)$ , mismo que cumple con el siguiente espacio de estados:

$$x = x_1; \tag{3.48}$$

$$x_2 = \dot{x}_1; \tag{3.49}$$

$$\theta = x_3; \tag{3.50}$$

$$\dot{x}_3 = x_4; \tag{3.51}$$

$$\dot{x}_4 = -33.94x_3 - 5x_4 + 87.46x_{10} + f; \tag{3.52}$$

$$\dot{x}_{10} = -cc_t x_2 - (c + c_t)x_{10} - (p_x) \text{sing}(s_x); \tag{3.53}$$

$$x_{10} = \text{sat}(x_{10}); \tag{3.54}$$

donde  $s_x$  está dado por

$$s_x = \dot{\sigma}_x + c_t \sigma_x; \quad (3.55)$$

$$\sigma_x = x_2 + c(x_1 - x_{1d}); \quad (3.56)$$

$$y = x_5; \quad (3.57)$$

$$x_6 = \dot{x}_5; \quad (3.58)$$

$$\phi = x_7; \quad (3.59)$$

$$\dot{x}_7 = x_8; \quad (3.60)$$

$$\dot{x}_8 = -23.31x_7 - 3.2x_8 + 61.64x_9 + f; \quad (3.61)$$

$$\dot{x}_9 = -c_y c_{ty} x_6 - (c_y + c_{ty}) x_9 - (p_y) \text{sing}(s_y); \quad (3.62)$$

$$x_9 = \text{sat}(x_9); \quad (3.63)$$

donde  $s_y$  está dada por

$$s_y = \dot{\sigma}_y + c_{ty} \sigma_y; \quad (3.64)$$

$$\sigma_y = x_6 + c_y(x_5 - x_{5d}); \quad (3.65)$$

con la función signo aproximada a:

$$\text{sing}(s) \approx \frac{s}{s + \zeta}, \zeta \rightarrow 0 \quad (3.66)$$

Las Figuras 3.16 y 3.17 presentan las respuestas y señales de control en ausencia de perturbaciones, mientras que las Figuras 3.18 y 3.19 en presencia de éstas. En la Tabla 3.4 se pueden observar los índices de desempeño de esta estrategia de control en ausencia de perturbaciones, para las Figuras 3.17 y 3.19 la señal azul representan la señal de control, mientras que la naranja la superficie deslizante.

Ante presencia de una perturbación en 15 segundos, con una amplitud de 2.5 veces el peso del móvil y una frecuencia de oscilación de 10 Hz, las posiciones translacionales y las posiciones angular  $\phi$  y  $\theta$  se muestran en 3.18, donde se observan respuestas rápidas y con un excelente rechazo a perturbación. Sus respectivas señales de control se presentan en la Figura 3.19, donde se puede

observar que ante la presencia de la perturbación el castaño de la señal de control creció de forma acotada.

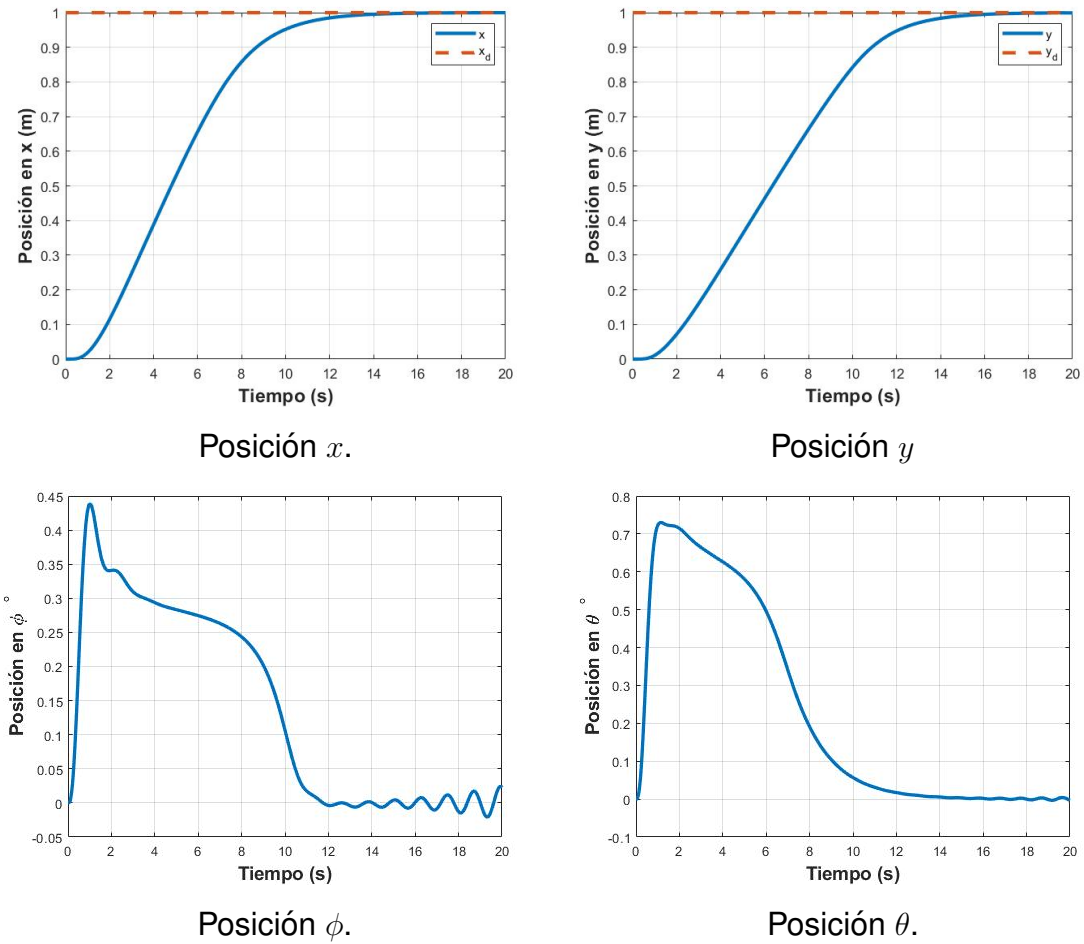
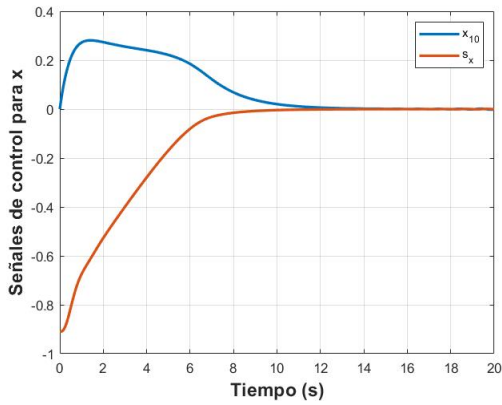
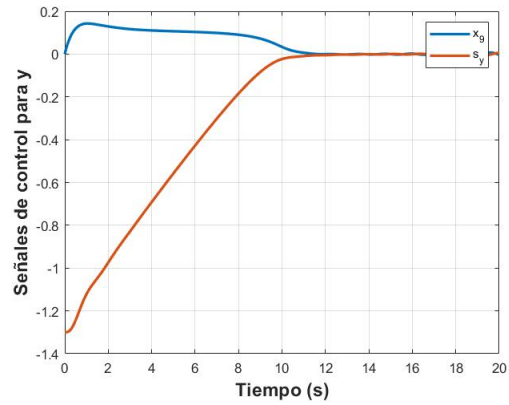


Figura 3.16: Posiciones del Cuasi-CMD asintótico.



Señal de control para  $x$ .



Señal de control para  $y$ .

Figura 3.17: Señales de control del Cuasi-CMD asintótico.

En la Tabla 3.5 se pueden observar los índices de desempeño de esta estrategia de control en presencia de una perturbación.

Tabla 3.5: Índices de desempeño en cuasi CMD asintótico.

Índice	$x$	$y$
IEC	3.6091	4.6864
IVAE	5.1101	6.5531
ITVAE	16.6853	26.8775
ITEC	8.2564	14.0020



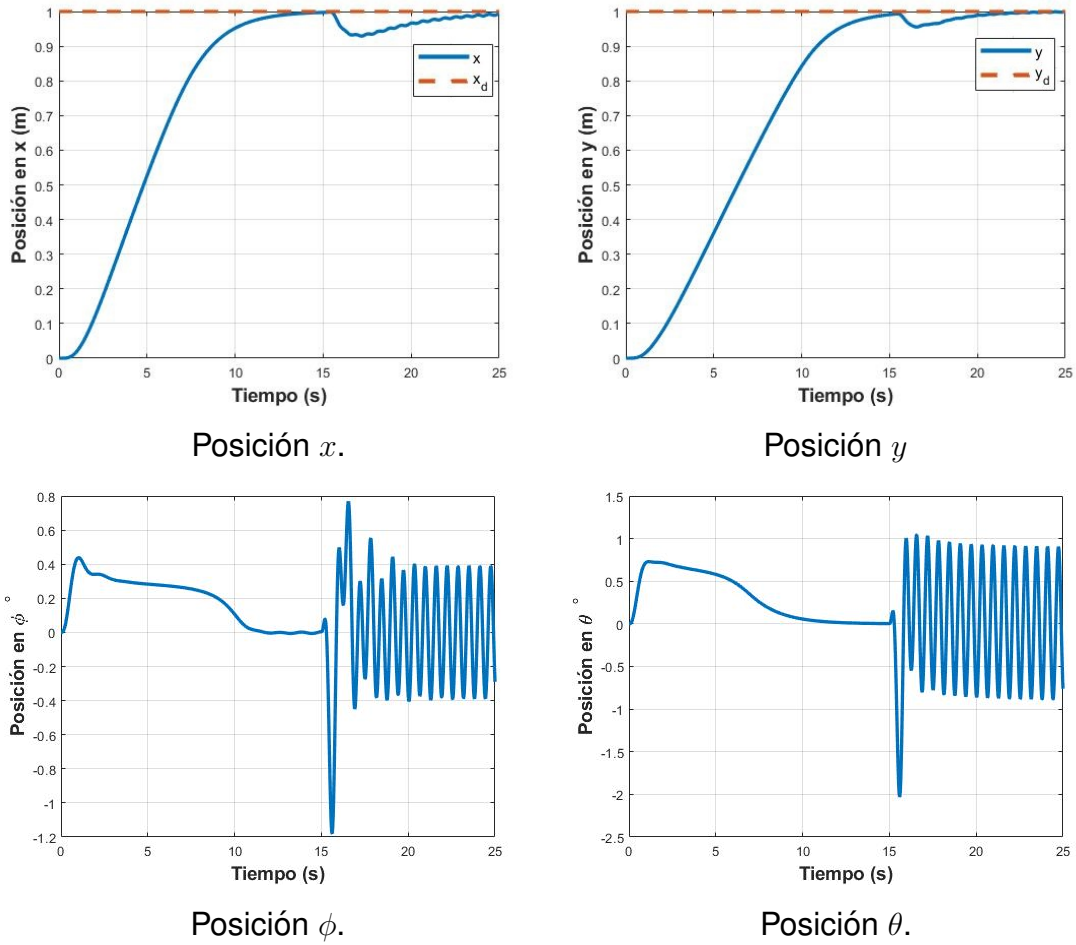


Figura 3.18: Posiciones del Cuasi-CMD asintótico ante perturbación.

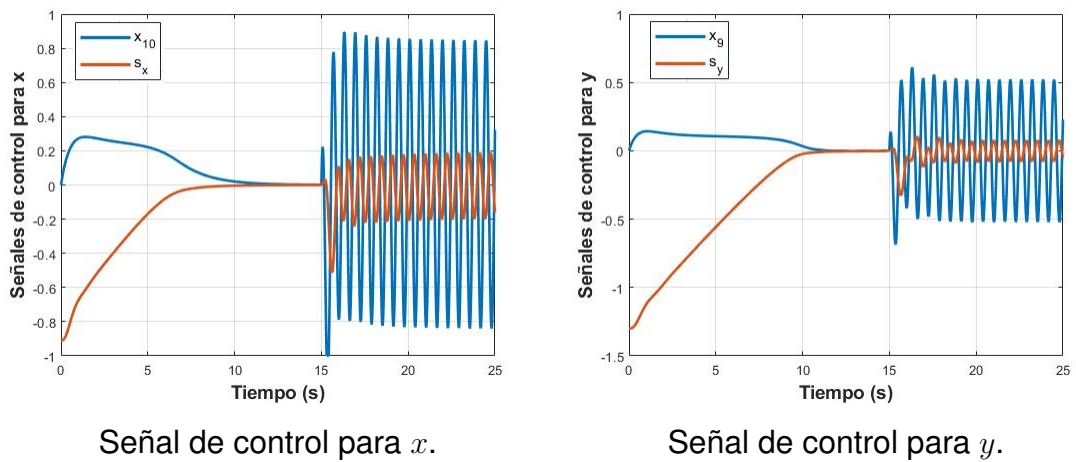


Figura 3.19: Señales de control del Cuasi-CMD asintótico ante perturbación.

### **3.3. Conclusión**

A través de las simulaciones presentadas se demuestra la posibilidad de implementar controladores clásicos en cascada, como el PID, en un cuadirrotor pese a que se trate de un sistema no lineal y que se un sistema que pueda presentar indeterminaciones. Esto siempre y cuando se tenga el cuidado de operar en las regiones en las que se encuentra completamente definido el modelo.

Respecto al controlador no lineal se seleccionó el CMD debido a la robustez que este presenta ante perturbaciones y dinámicas no modeladas, lo cual es una gran ventaja respecto a los controladores clásicos, pero el uso de la función signo, aun en sus aproximaciones, añade una fuerte carga extra de costo computacional.

# CAPÍTULO 4

RESULTADOS  
EXPERIMENTALES.

# Capítulo 4

## Resultados experimentales.

El objetivo principal de este capítulo, es presentar una descripción y metodología de los sistemas de captura de movimientos: sistema de radiofrecuencia y basado en visión artificial, así como de la plataforma del VANT que fueron validados.

También se muestran los resultados experimentales de los controladores descritos en el capítulo 3, el control PID y el control por modos deslizantes aplicados en tiempo real a la plataforma comercial AR DRone 2.0 de la marca Parrot.

La sección 4.1 muestra las plataformas de monitoreo implementadas, iniciando por el sistema comercial basado en radiofrecuencia y seguido de la paquetería de SLAM Visual ORB SLAM y terminando con TUM AR Drone, un proyecto de código abierto que combina la paquetería de SLAM visual PTAM y el controlador para AR Drone en ROS. La sección 4.2 describe los resultados experimentales de las estrategias de control validadas en tiempo real, las cuales son: a) controlador PD, b) controlador P en cascada y c) Controlador por modos deslizantes. Finalmente en la sección 4.3 se presentan las conclusiones de este capítulo.

### 4.1. Plataformas de monitoreo.

En este tema de tesis se probaron dos sistemas de monitoreo, el primero es un sistema comercial por radiofrecuencia llamado “GPS indoor, start set HW v4.9” y el segundo es un sistema que se realizó en base a visión artificial utilizando SLAM, PTAM, Ardrone autonomy y ROS.

### 4.1.1 Sistema por radiofrecuencia.

Se trabajó en un plataforma de monitoreo basada en el sistema de posicionamiento por radio frecuencia “GPS indoor, start set HW v4.9”, este sistema consta de al menos 2 balizas fijas que se montan en puntos de referencia, una móvil y un módem con conectividad USB.

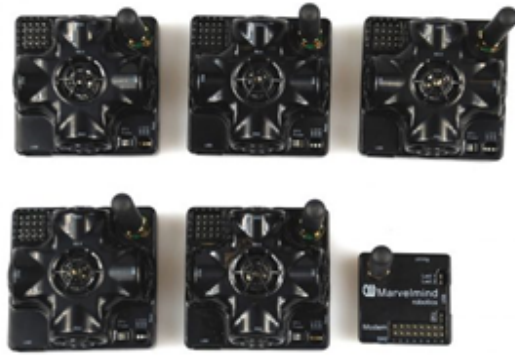
La baliza móvil se monta sobre el cuerpo del cual se desea conocer la ubicación (relativa a las balizas fijas), además cuenta con una unidad de medición inercial (IMU por sus siglas en inglés), mediante la cual se pueden obtener una información más detallada del estado de movimiento de la baliza.

Se realizó un experimento para calcular el error promedio en la estimación de la ubicación, donde en un área de  $10m^2$  con 3 balizas fijas, se tomaron 20 mediciones cada una a  $20cm$  de la anterior, obteniéndose un error promedio de  $1.3cm$  y una frecuencia máxima de actualización de  $7.5Hz$ , en la Figura 4.1 se muestran los datos del experimento.

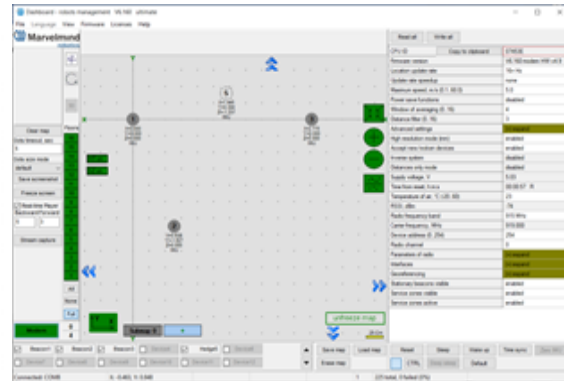
Iteración	X0	Y0	Z0	X1	Y1	Z1	Cal	Real	%
1	-2.793	-1.385	-1.374	-2.778	-1.213	-1.377	0.172679	0.2	13.66
2	-2.793	-1.385	-1.374	-2.791	-0.975	-1.363	0.410152	0.4	2.54
3	-2.793	-1.385	-1.374	-2.79	-0.576	-1.328	0.810312	0.8	1.29
4	-2.793	-1.385	-1.374	-2.789	-0.332	-1.318	1.054496	1	5.45
5	-2.793	-1.385	-1.374	-2.79	-0.331	-1.325	1.055143	1	5.51
6	-2.793	-1.385	-1.374	-3.149	-1.366	-1.384	0.356647	0.4	10.84
7	-2.793	-1.385	-1.374	-3.08	-1.37	-1.39	0.287837	0.3	4.05
8	-2.793	-1.385	-1.374	-2.98	-1.382	-1.4	0.188823	0.2	5.59
9	-2.814	-1.37	-1.385	-2.91	-1.381	-1.372	0.097499	0.1	2.50
10	-1.506	-1.629	-1.428	-1.406	-1.619	-1.44	0.101213	0.1	1.21
11	-1.506	-1.629	-1.428	-1.301	-1.624	-1.461	0.207699	0.2	3.85
12	-1.506	-1.629	-1.428	-1.2	-1.612	-1.453	0.30749	0.3	2.50
13	-1.506	-1.629	-1.428	-1.06	-1.69	-1.32	0.462927	0.4	15.73
14	-1.506	-1.629	-1.428	-0.967	-1.68	-1.37	0.544505	0.5	8.90
15	-1.506	-1.629	-1.428	-1.177	-1.769	-1.831	0.538749	0.6	10.21
16	-1.506	-1.629	-1.428	-1.126	-1.7	-1.906	0.614756	0.7	12.18
17	-1.506	-1.629	-1.428	-1.508	-1.742	-1.435	0.113234	0.1	13.23
18	-1.506	-1.629	-1.428	-1.517	-1.823	-1.472	0.199231	0.2	0.38
19	-1.506	-1.629	-1.428	-1.553	-1.92	-1.504	0.304411	0.3	1.47
20	-0.685	-3.677	-1.556	-0.665	-3.614	-1.501	0.085988	0.1	14.01
								PROM.	6.76

Figura 4.1: Error en mediciones GPS Indoor.

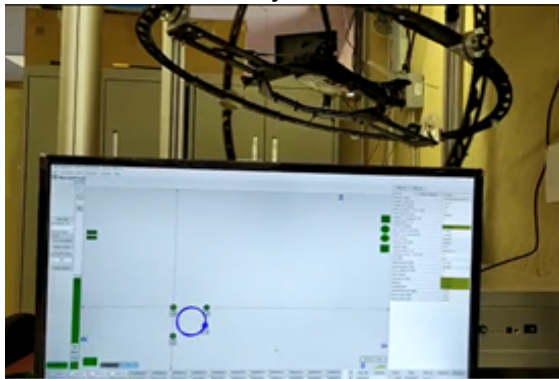
Se comunicó el conjunto GPS Indoor con la plataforma de desarrollo libre Arduino mediante comunicación serial (véase Figura 4.2). Con base a dicha plataforma se programó un sistema en Matlab que monitorea el estado del cuadrirrotor en tiempo real. En la Figura 4.3 se muestra los planos  $XY$ ,  $ZY$ ,  $ZX$ , y a  $x$ ,  $y$  y  $z$  en relación al tiempo  $t$ .



Balizas y módem.



Aplicación del fabricante.



Señal de control para  $x$ .



Seguimiento del VANT en vuelo.

Figura 4.2: GPS Indoor HW v4.9.

En la Figura 4.4 se muestra ejemplos de la comunicación del módulo con Arduino, donde se despliega la información en una pantalla LCD, esto se realizó para validar la información desplegada en la aplicación en Matlab. Se muestra a) la comunicación del módem con Arduino y b) la comunicación de una baliza con Arduino. Con lo que se corrobora que la información se puede obtener directamente de la baliza para el caso que se conecte directa a una computadora de bolsillo (por ejemplo Raspberry Pi) y se valida la opción de tener una estación de control desde donde se monitoree el estado dinámico del móvil.

Además se validó la posibilidad de utilizarlo en Linux Ubuntu 16.04 con el entorno

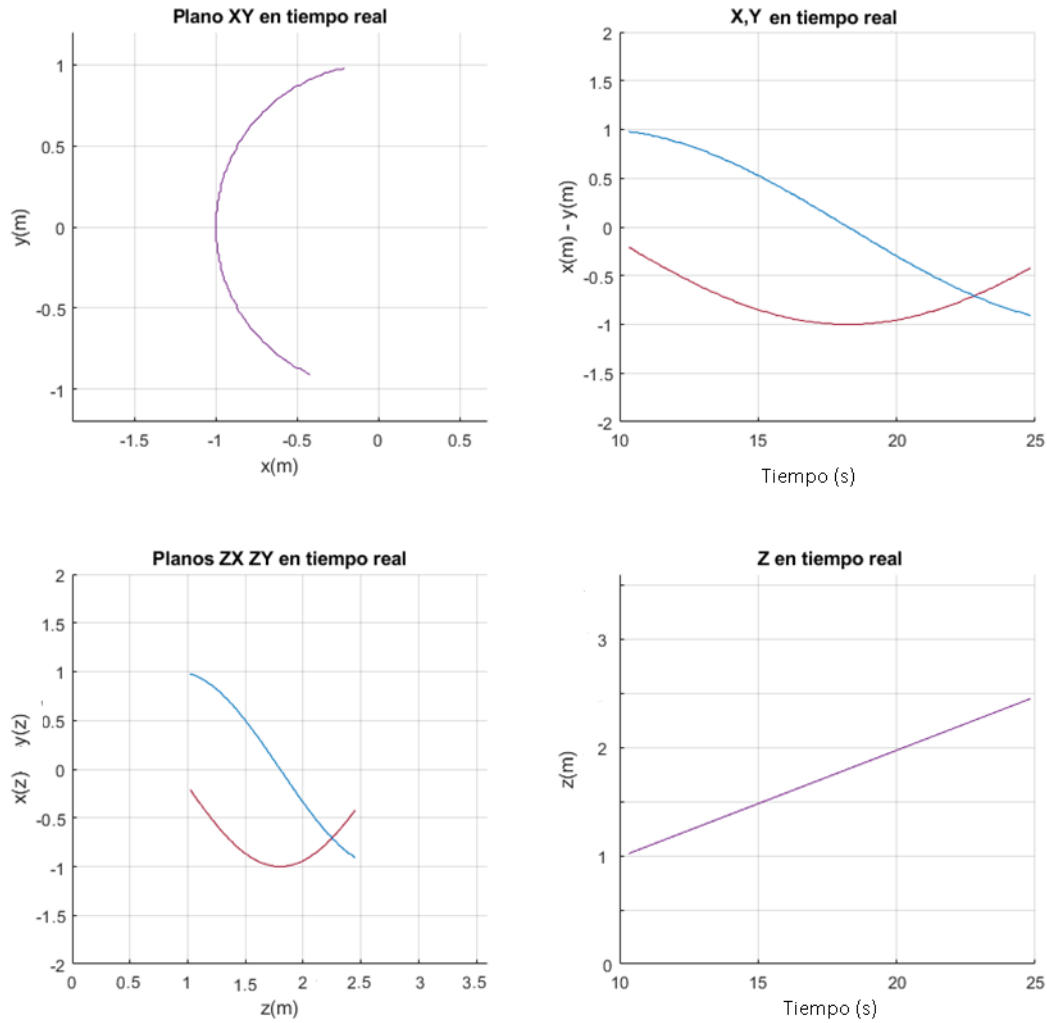
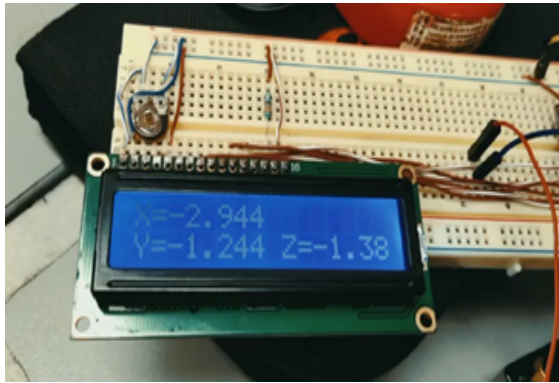
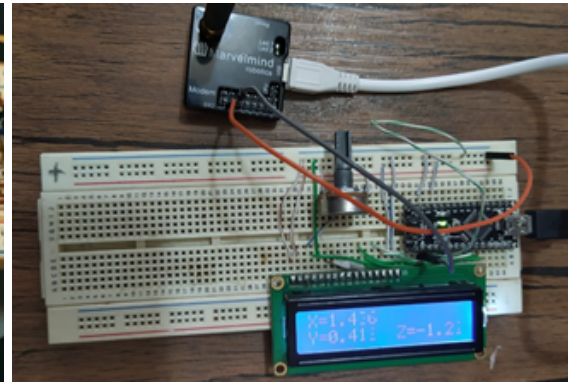


Figura 4.3: Monitoreo en tiempo real en Matlab.

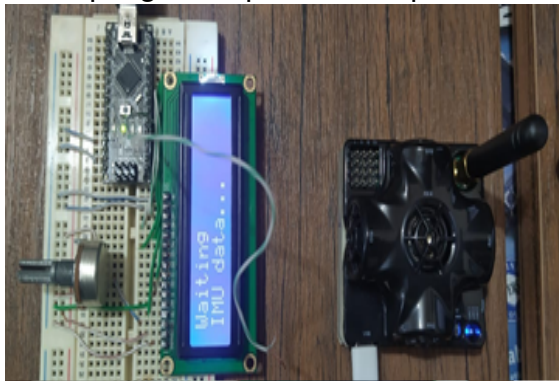
de ROS, y de esa manera utilizarlo como medio de retroalimentación para un futuro sistema en lazo cerrado con el vehículo AR. Drone 2.0. La Figura 4.5 muestra en la posición publicada en la terminal y la posición mostrada en un mapa en la aplicación RVIZ.



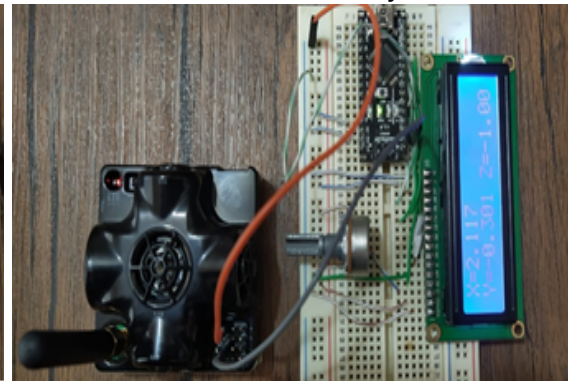
Despliegue en pantalla de posición.



Comunicación Arduino y Modem.

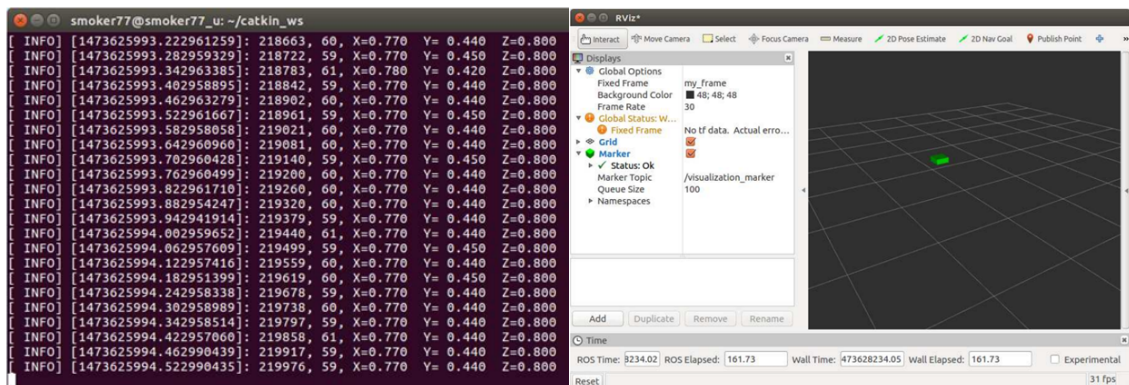


Comunicación Arduino y baliza.



Comunicación Arduino y baliza II.

Figura 4.4: Comunicación Arduino y GPS Indoor HW v4.9.



Posición publicada.

Posición en un mapa.

Figura 4.5: GPS Indoor en ROS.



### 4.1.2 Visión artificial.

Una forma para obtener la retroalimentación de los estados es a través de sistemas de visión en arquitecturas de código abierto, en general estos podrían subdividirse en dos clases o tópicos principales, los sistemas de odometría visual y los sistemas de mapeo y localización simultáneos (SLAM). SLAM es un tópico importante de investigación a lo largo de las últimas dos décadas en campo de la visión computarizada y la robótica.

Las técnicas de SLAM construyen un mapa de un entorno desconocido y localizan al sensor dentro del mapa haciendo especial hincapié en su operación en tiempo real [46]. Se utilizan distintos sensores, pero actualmente el SLAM basado en visión tiene especial interés. Visual SLAM puede realizarse utilizando una cámara sencilla (monocular), cámaras estéreo, cámaras RGB-D, combinándose además de sensores tales como ultrasónicos y unidades inerciales. ORB-SLAM es un sistema de visión basado en características y no en filtros, de manera particular está basado en “Oriented FAST and rotated BRIEF (ORB) que es la fusión de un detector rápido de puntos clave y un breve descriptor.

A la fecha de este documento de tesis se han lanzado 2 actualizaciones de ORB-SLAM, teniendo así ORB-SLAM, ORB-SLAM2 y OB-SLAM 3. ORB-SLAM 3 (el cual se utiliza en este trabajo) cuenta con una excelente posición y robustez, las asociaciones de los datos las realiza a partir de descriptores, utiliza la paquete de palabras DBoW2, su estimación se realiza de manera local a partir del ajuste de paquetes (BA), y realiza en cierre de lazo a partir de la bolsa de palabras y del BA. Pueden utilizarse con cámaras monoculares, estéreo y ojo de pez. Además, puede fusionarse con unidades de medición inercial (IMU) y permite la utilización de multi-mapas, para más información al respecto de las técnicas de SLAM y odometría visual se recomienda leer [47], en el se encuentra un resumen del estado del arte de estas técnicas. En la Figura 4.6 se muestra a grandes rangos el proceso seguido por ORB-SLAM en las tareas de mapeo, localización y lazo cerrado.

La Figura 4.7 muestra la organización del SDK del control del AR Drone, la cual consta de 3 canales de comunicación wi-fi mediante puertos UDP. Uno de estos tres, se encuentra dedicado a la transmisión de vídeo, siendo posible seleccionar si el vídeo es transmitido desde la cámara frontal o bien desde la cama inferior. Otro

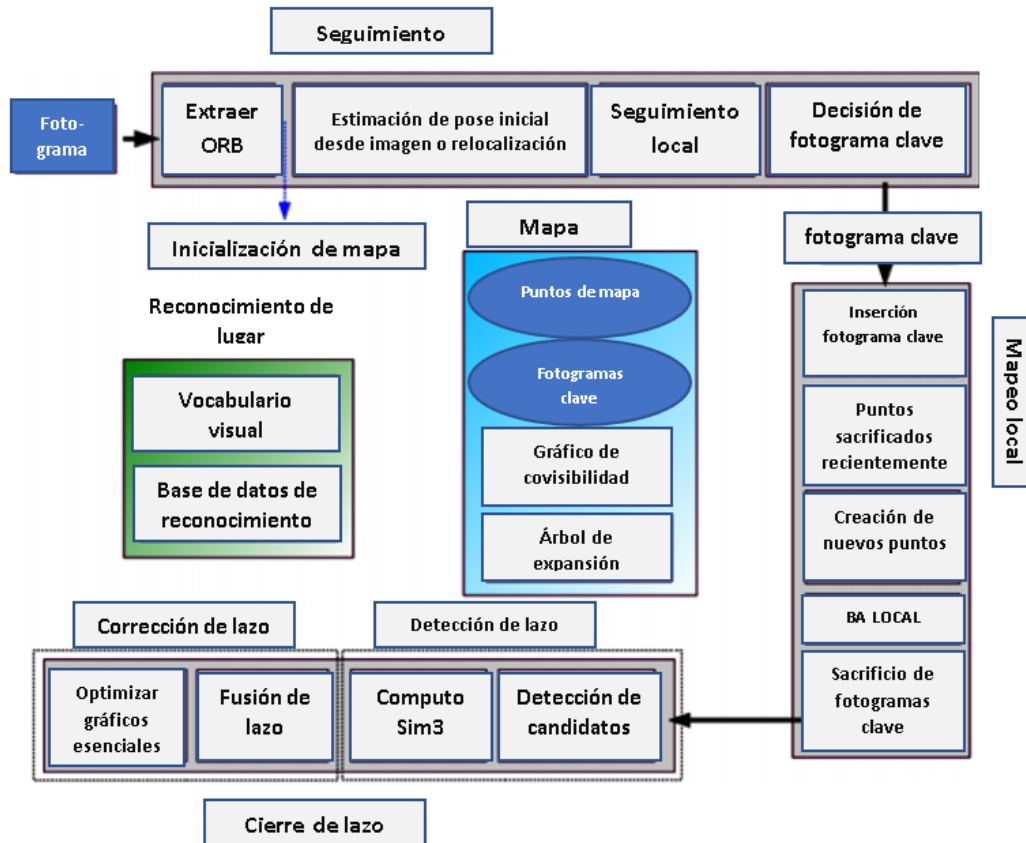


Figura 4.6: ORB-SLAM.

de los puertos, se encuentra dedicado al control y comunicación a través del envío de comandos en formato AT y el tercer puerto, se encuentra dedicado a la retroalimentación de la información de vuelo, es decir, el estado, la posición, la velocidad y la velocidad de rotación de los motores.

Ardrone autonomy es un controlador desarrollado en el entorno de ROS por Autonomy Lab de la universidad Simon Fraser, con base al SDK previamente mencionado, mediante el cual se puede manipular tanto al AR Drone 1.0 como AR Drone 2.0. Su primera publicación fue en julio del 2012 y su última actualización en abril del 2014. Dado a que su actualización más reciente data del 2014, cuenta respaldo solamente hasta la distribución Kinetic de ROS lanzada en 2016 y vigente hasta 2021.

Ardrone autonomy genera tópicos tanto para publicar comandos al cuadricóptero, como para suscribirse a ellos y obtener información de éste. La información recibida

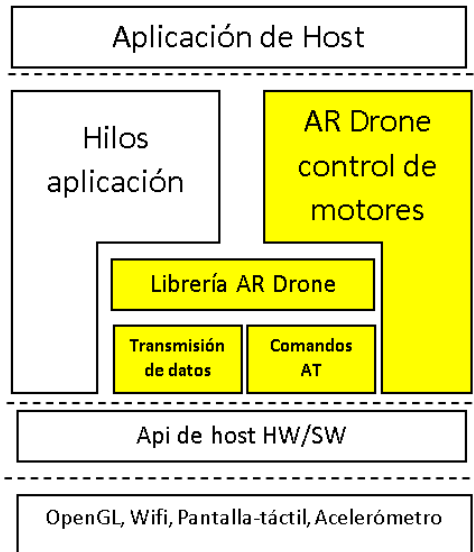


Figura 4.7: Layout del SDK.

se publica en el tópico ardrone/navdata y contiene la siguiente información:

- header: Cabecera del mensaje de ROS.
- batteryPercent: Porcentaje de batería.
- state: Estado del Drone, mismo que puede ser:
  - 0: Unknown.
  - 1: Initied.
  - 2: Landed.
  - 3,7: Flying.
  - 4: Hovering.
  - 5: Test (?).
  - 6: Taking off.
  - 8: Landing.
  - 9: Looping (?).
- rotX: Left/right: Ángulo de inclinación en grados al rededor de  $x$ .
- rotY: Forward/backward Ángulo de inclinación en grados al rededor de  $y$ .

- rotZ: Ángulo de inclinación en grados al rededor de  $z$ .
- magX, magY, magZ: Lecturas del Magnetómetro.
- pressure: Presión (Pa).
- temp : Temperatura.
- windspeed.
- windangle.
- wind-comp-angle.
- altd: Altura estimada (mm).
- motor1..4: Motor VALOR PWM.
- vx, vy, vz: Velocidad lineal (mm/s) [TBA: Convención].
- ax, ay, az: aceleración (g) [TBA: Convención].
- tm: Estampa de tiempo.

Este controlador creará tres tópicos para la transmisión de vídeo de cada cámara: ardrone/image-ra, ardrone/front/image-raw y ardrone/bottom/image-raw. Cada uno de estos tres son interfaces de cámara ROS estándar y publican mensajes de tipo transporte de imágenes. El móvil despegará, aterrizará o realizará un paro de emergencia / reiniciará de emergencia si se publica un mensaje en los tópicos ardrone/takeoff, ardrone/land y ardrone/reset respectivamente.

Para volar el móvil después del despegue, puede publicar un mensaje del tópico cmd-vel, el rango para cada componente debe estar entre -1.0 y 1.0:

- -linear.x: moverse hacia atrás.
- + linear.x: avanzar.
- -linear.y: moverse a la derecha.
- + linear.y: moverse a la izquierda.

- -linear.z: moverse hacia abajo.
- + linear.z: subir.
- -angular.z: gira a la derecha.
- + angular.z: girar a la izquierda.

Manipulando los argumentos del tópico `cmd-vel`, el cuadrirrotor se desplazará a la posición angular deseada, o bien a la velocidad de traslación vertical y angular al rededor de  $z$ , siendo este un lazo abierto para la posición espacial  $(x, y, z)$ . En la Figura 4.8 se muestra la relación de los tópicos necesarios para realizar un control en lazo abierto.

ORB-SLAM3 no es compatible con la distribución mencionada de ROS, para utilizarlo es necesario cambiar las versiones de los compiladores. En este trabajo se utilizan los tópicos `ardrone/front/image-raw` y `ardrone/imu` para implementar ORB-SLAM3.

Una vez instalado ORB-SLAM3, en sus ejemplos de ROS se encuentra el archivo `ros-mono`, para ejecutarlo se necesita proporcionarle tres argumentos: *a*) vocabulario, se utiliza el que se proporciona por defecto. *b*) tópico en el cual se encuentra la transmisión de vídeo e IMU, y *c*) configuración de la cámara. La cámara se encuentra girada  $-90, 0, -90$  grados y a una distancia de  $0.21$  m del centro de masa. A continuación se muestran las configuraciones de la cámara frontal de AR Drone 2.0.

- `Camera.fx: 564.6175815337156.`
- `Camera.p2: 0.01744527789230373.`
- `Camera.fy: 540.8785822897302.`
- `Camera.k3: 0.`
- `Camera.cx: 298.690396701664.`
- `Camera.fps: 30.0.`
- `Camera.cy: 235.1317331683574.`
- `Camera.RGB: 1.`
- `Camera.k1: -0.5422234679833581.`
- `ORBextractor.nFeatures: 5000.`
- `Camera.k2: 0.3414935368470705.`
- `ORBextractor.scaleFactor: 1.2.`
- `Camera.p1: -0.05892040284098003.`
- `ORBextractor.nLevels: 8.`

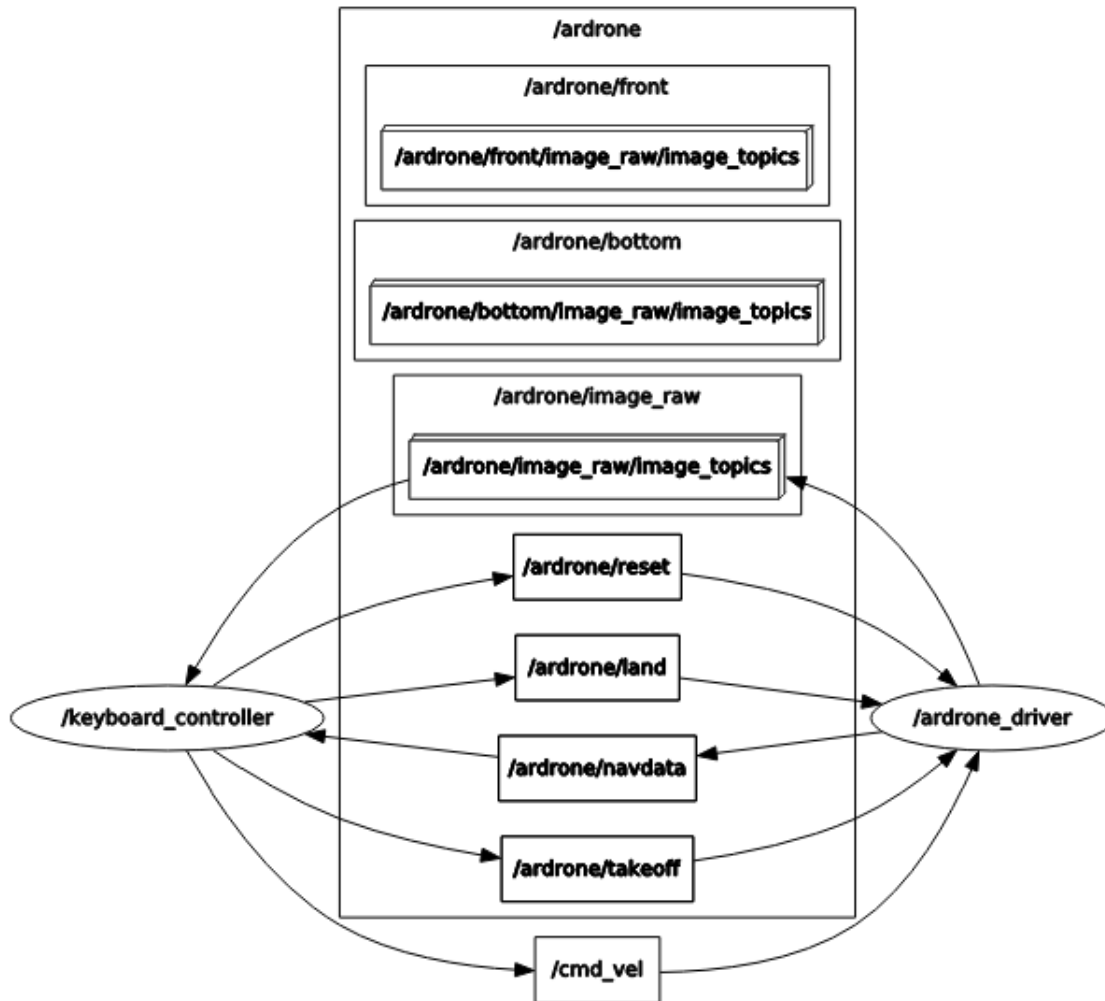


Figura 4.8: AR Drone y sus tópicos en ROS.

- ORBextractor.iniThFAST: 20.
- ORBextractor.minThFAST: 7.
- Viewer.KeyFrameSize: 0.05.
- Viewer.KeyFrameLineWidth: 1.
- Viewer.GraphLineWidth: 0.9.
- Viewer.PointSize:5.
- Viewer.CameraSize: 0.08.
- Viewer.CameraLineWidth: 3.
- Viewer.ViewpointX: 0.
- Viewer.ViewpointY: -0.7.
- Viewer.ViewpointZ: -1.8.
- Viewer.ViewpointF: 500.

En la Figura 4.9 se visualiza una toma de la cámara en tiempo real, esta misma

toma habiendo sido procesada por ORB SLAM y extraídos los puntos clave, el mapa generado por la extracción sucesiva de puntos clave de distintas tomas, y por último el AR Drone en vuelo.



Imagen capturada por la cámara frontal.

Imagen procesada por ORB SLAM.

Mapa generado por ORB SLAM.

AR Drone en vuelo.

Figura 4.9: Implementación de ORB SLAM en AR Drone.

Si ya se cuenta con ORB SLAM 3 funcionando es posible obtener la transmisión de vídeo desde una Raspberry Pi, siempre que ésta se configure como esclavo en ROS, ejecutando sobre ésta el controlador usb-cam, es necesario fijar la cámara y generar una matriz homogénea con su trama respecto a donde se encuentre el sensor IMU, de esta manera la Raspberry Pi se convierte en un sensor de movimiento inteligente, lo anterior se realizó y se muestra en la Figura 4.10.

### 4.1.3 TUM AR Drone.

TUM AR Drone es una paquetería de ROS en la que se implementa PTAM para el control y monitoreo del AR Drone. El sistema permite controlar al cuadrirrotor desde una computadora, consta de tres componentes: un sistema SLAM

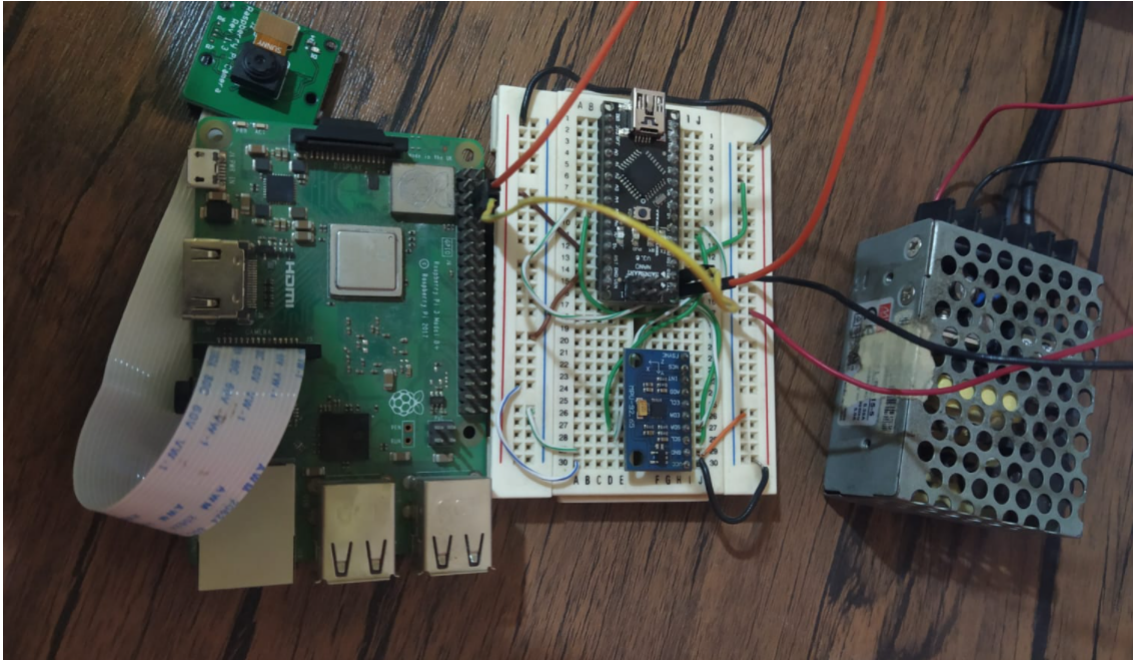


Figura 4.10: Raspberry Pi y ORB SLAM.

monocular, un filtro Kalman extendido para la fusión de datos y la estimación del estado y un controlador PID para generar comandos de dirección. A través de esta paquetería el cuadricóptero es capaz de navegar en entornos previamente desconocidos sin requerir marcadores o sensores externos. Además, el sistema cuenta su robustez a la pérdida temporal de seguimiento visual y retrasos significativos en el proceso de comunicación.

La paquetería genera los siguientes tres nodos:

- `/drone-stateestimation`: Estimación del estado de los drones, incluidos PTAM y visualización.
- `/drone-autopilot`: Controlador del drone, requiere la estimación de los estados del drone.
- `/drone-gui`: GUI para controlar el drone (con un Joystick o teclado) y para controlar el piloto automático y el nodo de estimación de estado.

Nodo `drone-stateestimation`: Calcula la posición del móvil según los datos de navegación enviados, los comandos de control enviados y PTAM. Tópicos suscritos:

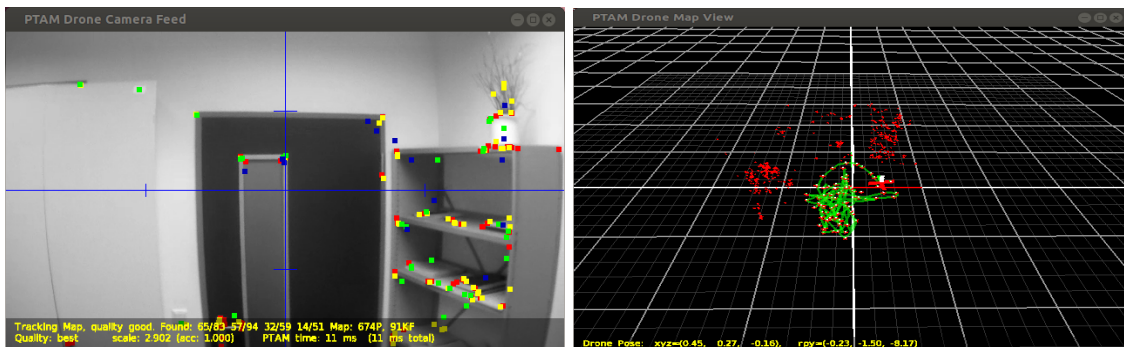


- /ardrone / navdata.
- /ardrone / image-raw.
- /cmd-vel.
- /tum-ardrone / com.

Tópicos publicados:

- /ardrone / predictedPose.
- /tum-ardrone / com.

En la Figura 4.11 se presentan un ejemplo del funcionamiento en tiempo real la paquetería, de la imagen en tiempo real se extraen los puntos clave y se vierten sobre el mapa 3D donde se reflejando los puntos como la pose el móvil.



Vídeo y los puntos del mapa de PTAM.

Mapa PTAM.

Figura 4.11: Ventanas de TUM AR Drone.

Nodo drone-gui: este nodo ofrece una GUI QT simple para controlar el nodo drone-autopilot, el nodo drone-stateestimation y volar el móvil manualmente mediante el teclado o el joystick.

Tópicos suscritos

- /cmd-vel.
- /tum-ardrone / com.
- /ardrone / takeoff.

- /ardrone / land.
- /ardrone / reset.
- /ardrone / predictedPose.
- /ardrone / navdata.
- /joy.

#### Tópicos publicados

- /cmd-vel.
- /tum-ardrone / com.
- /ardrone / despegue.
- /ardrone / land.
- /ardrone / reset.

Nodo drone-autopilot: controlador PID para el móvil. También incluye seguimiento básico de puntos de ruta e inicialización automática. Requiere que se esté ejecutando drone-stateestimation. El objetivo se establece a través del tema tum-ardrone / com. Este nodo despliega la interfaz gráfica mediante la cual se controla el cuadricóptero, la cual se muestra en la Figura 4.12.

Tópicos suscritos: ardrone / predictedPose

Tópicos publicados

- / cmd-vel.
- / ardrone / land.
- / ardrone / takeoff.
- / ardrone / reset.

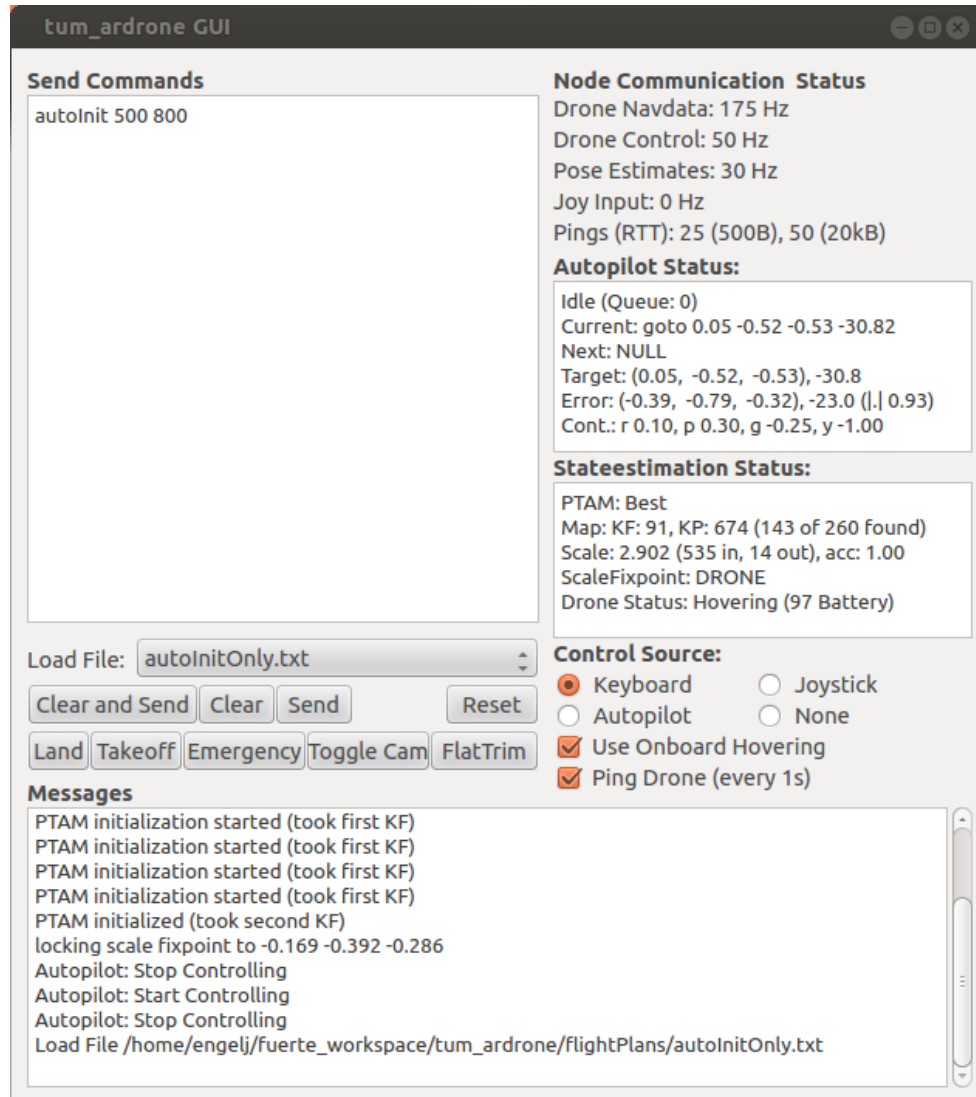


Figura 4.12: Interfaz de TUM AR Drone.

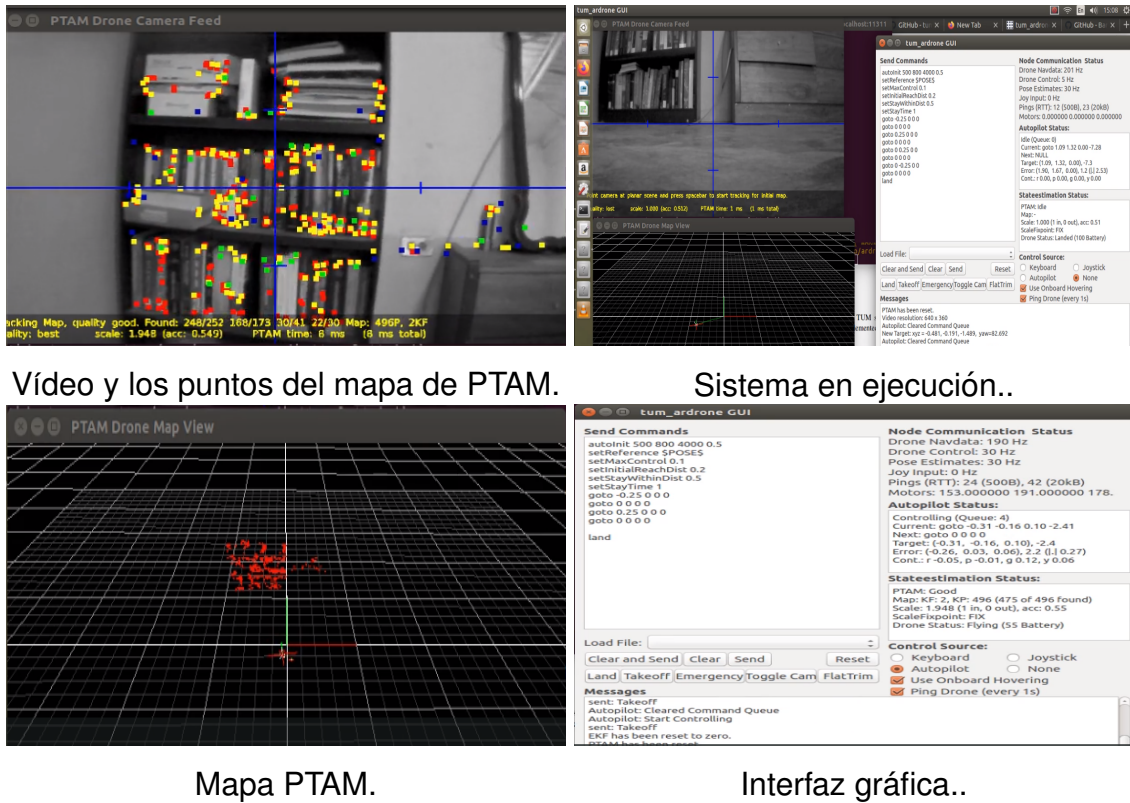


Figura 4.13: Ventanas de TUM AR Drone I.

## 4.2. Implementación en tiempo real.

### 4.2.1 Control convencional.

Para tener una referencia del comportamiento del control diseñado se voló el cuadricóptero en vuelo estacionario utilizando solo el control interno del mismo, en las Figuras 4.16 y 4.17 se muestran las gráficas del comportamiento en dos vuelos distintos. Como se puede observar no se encuentra completamente fijo, sino que genera oscilaciones, se configuró los parámetros de ángulo máximo de inclinación en 5 grados, velocidad vertical máxima en  $0.2 \text{ m/s}$  y velocidad máxima de rotación (yaw) En  $40 \text{ }^\circ/\text{s}$  de manera que esta oscilación no sea tan significativa. De esta manera se obtuvo un error máximo en  $z$  de 10 cm (referencia 75 cm), en  $x$  el sobre impulso fue de 15 cm, mientras que en  $y$  10 cm.

Con base a las identificaciones encontradas en la literatura se construyó e implemento un controlador con la estructura mostrada en la Figura 4.14. En una primera propuesta se seleccionó un control PD para el control de posición  $x$  y  $y$  y un control P para  $z$  y el ángulo  $\psi$ , dicha propuesta es descrita por las ecuaciones (4.1) - (4.4).

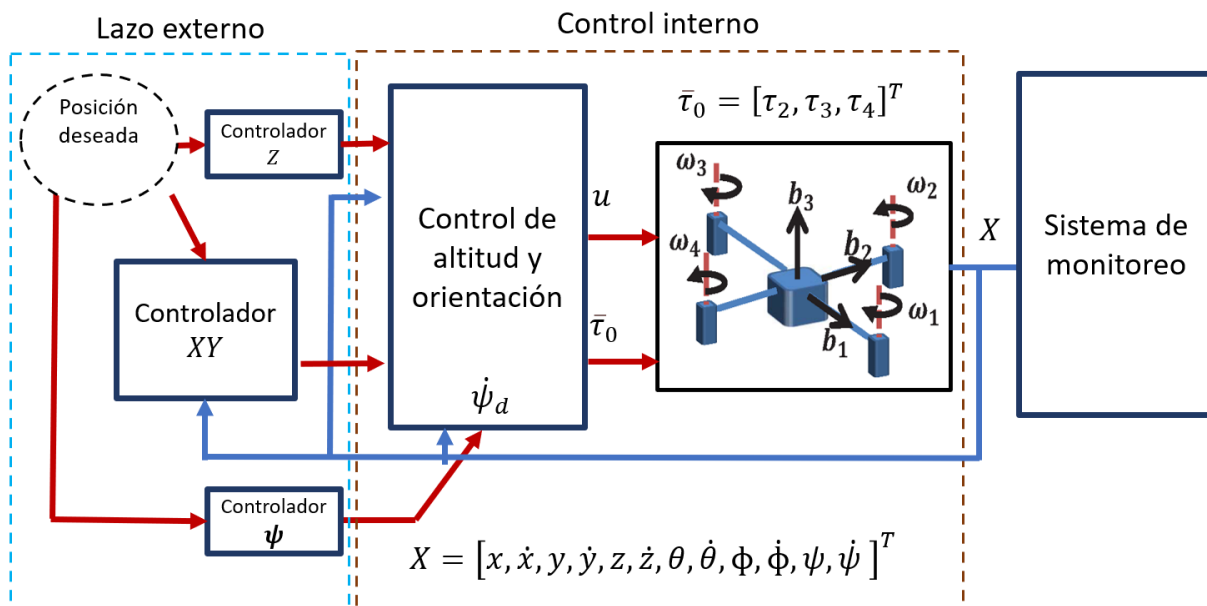


Figura 4.14: Esquema de control en tiempo real.

Las ecuaciones empleadas en esta implementación son las que siguen.

$$u_x = k_{px}(x_d - x) - k_{vx}\dot{x}. \quad (4.1)$$

$$u_y = k_{py}(y_d - y) - k_{vy}\dot{y}. \quad (4.2)$$

$$u_z = k_{pz}(z_d - z). \quad (4.3)$$

$$u_\psi = k_{p\psi}(\psi_d - \psi). \quad (4.4)$$

Como se puede observar en la Figura 4.15 donde se muestra el controlador PD en vuelo estacionario,  $z$  tiene un sobre impulso similar al del control interno, mientras que las posiciones  $x$  y  $y$  muestran un mejor comportamiento mostrando 8 y 9 cm de sobre impulso respectivamente.

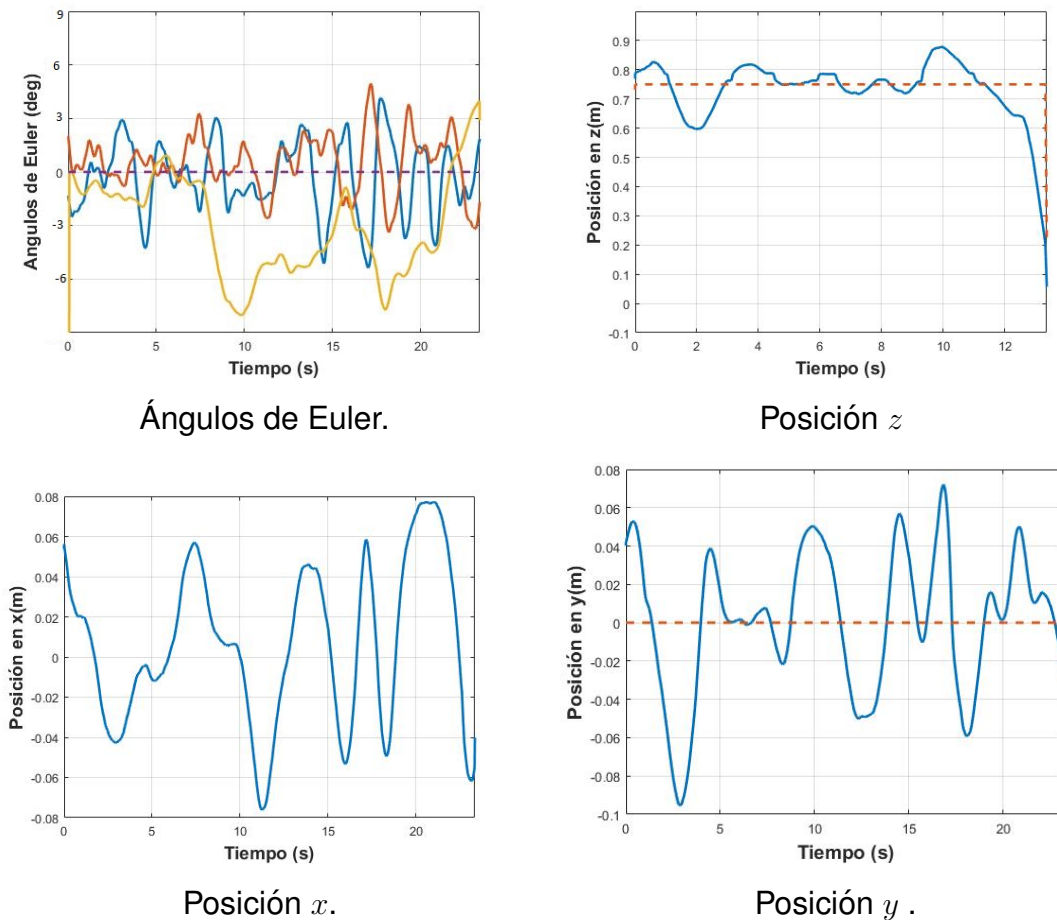


Figura 4.15: Resultados vuelos estacionarios con control PD.

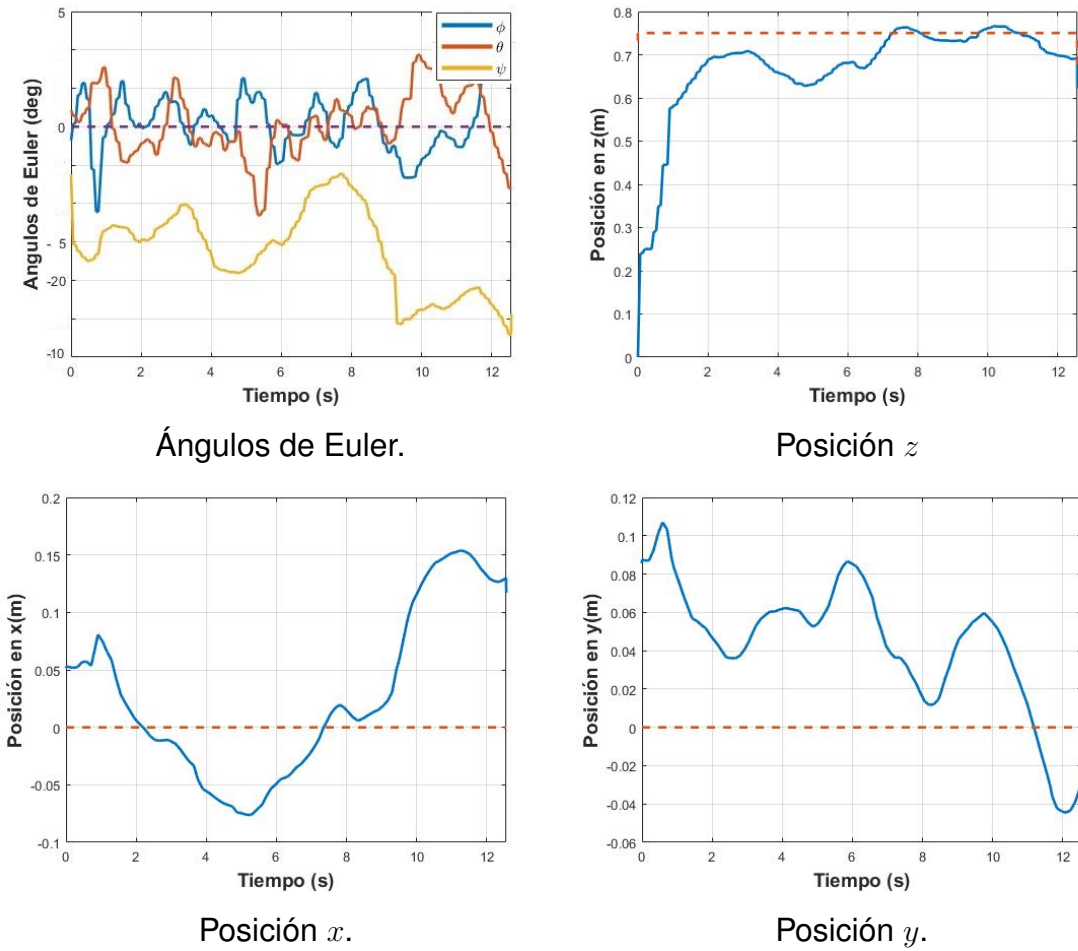


Figura 4.16: Resultados vuelos estacionarios con control interno I.

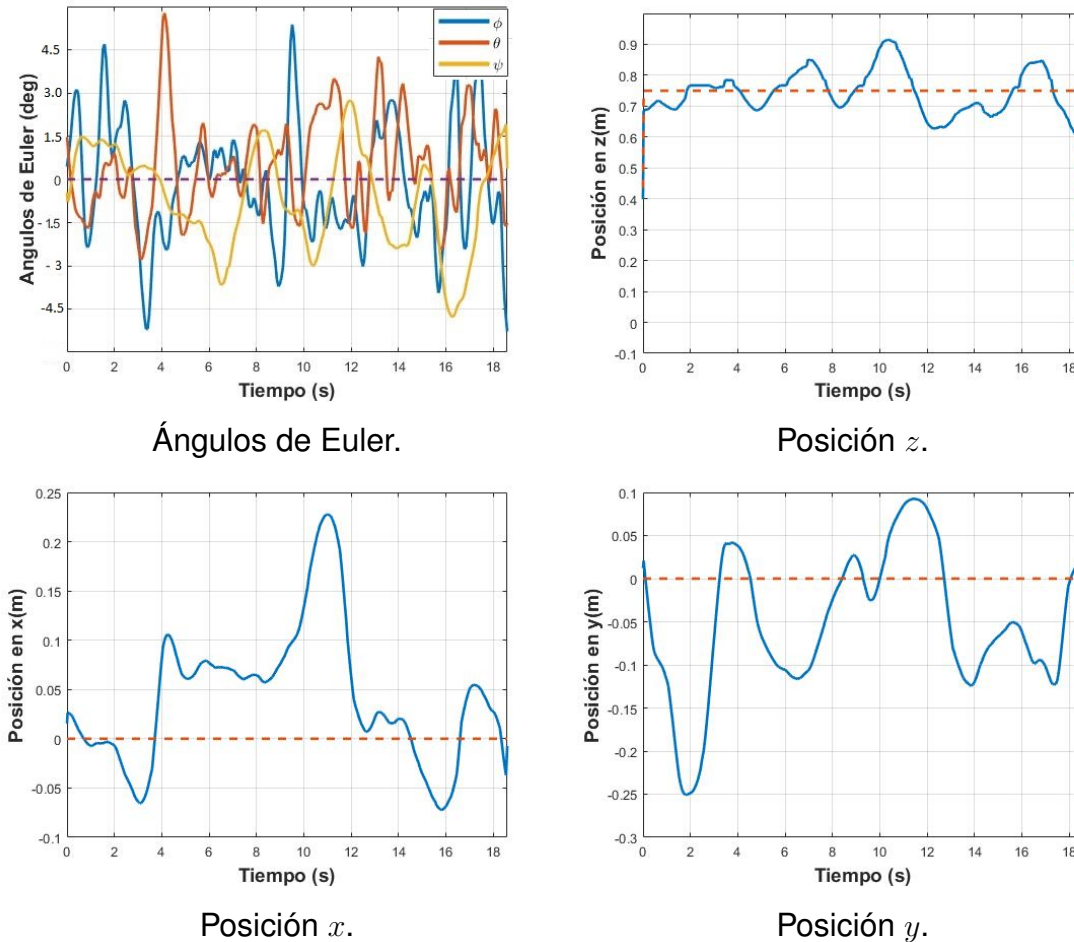


Figura 4.17: Resultados vuelos estacionarios con control interno II.

En la Figura 4.18 se muestra el resultado del control PD con referencias contantes distintas de cero. Como se puede observar el cuadricóptero responde adecuadamente a las estrada de control, convergiendo en las referencias solicitadas pese a las oscilaciones que ya presenta el móvil. Además, es posible visualizar el efecto del retardo en la comunicación.



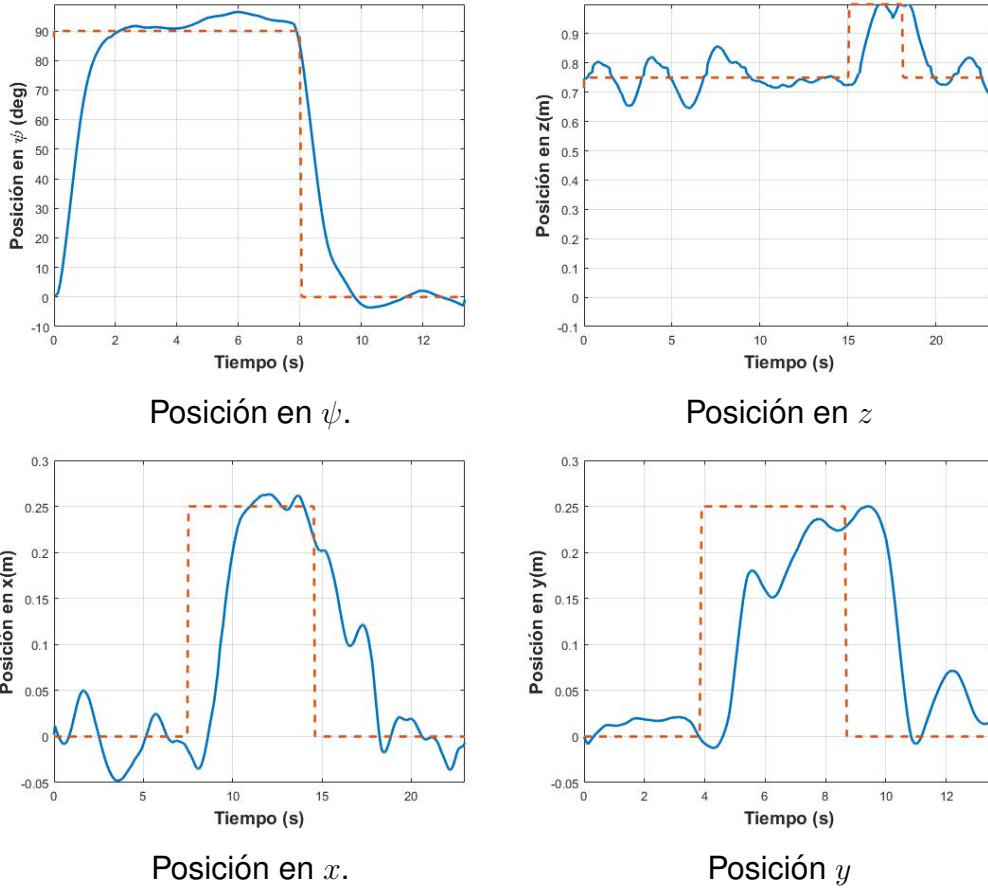


Figura 4.18: Resultados del lazo de control con referencias distintas a cero.

### 4.2.2 Control CMD.

Una segunda propuesta para el control de posición  $x$  y  $y$  es adecuar una función con modos deslizantes al control PD tal como se describe en las Ecuaciones (4.5) - (4.8). Como se observa en la Figura 4.19 el control también converge en la referencia pero con mayor oscilación y un sobre impulso de 2 cm en  $x$  y de 7 cm en  $y$ .

$$v_x = k_{px}(x_d - x) - k_{vx}\dot{x}. \quad (4.5)$$

$$u_x = v_x + p_x \text{sing}(v_x). \quad (4.6)$$

$$v_y = k_{py}(y_d - y) - k_{vy}\dot{y}. \quad (4.7)$$

$$u_y = v_y + p_y \text{sing}(v_y). \quad (4.8)$$

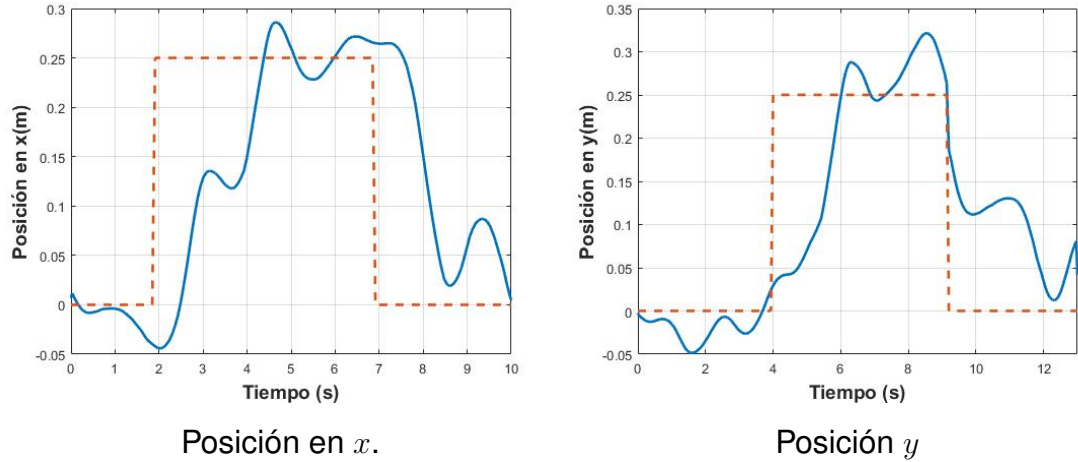


Figura 4.19: Control por modos deslizantes.

Además, se implementó también un lazo de control en cascada de posición y de velocidad acorde al esquema mostrado en en la Figura 4.20, descrito por las Ecuaciones (4.9) a (4.12).

$$v_{xd} = k_{px}(x_d - x), \quad (4.9)$$

$$\theta_d = -k_{pvx}(v_{xd} - v_x), \quad (4.10)$$

$$v_{yd} = k_{py}(y_d - y), \quad (4.11)$$

$$\phi_d = k_{pvy}(v_{yd} - v_y), \quad (4.12)$$

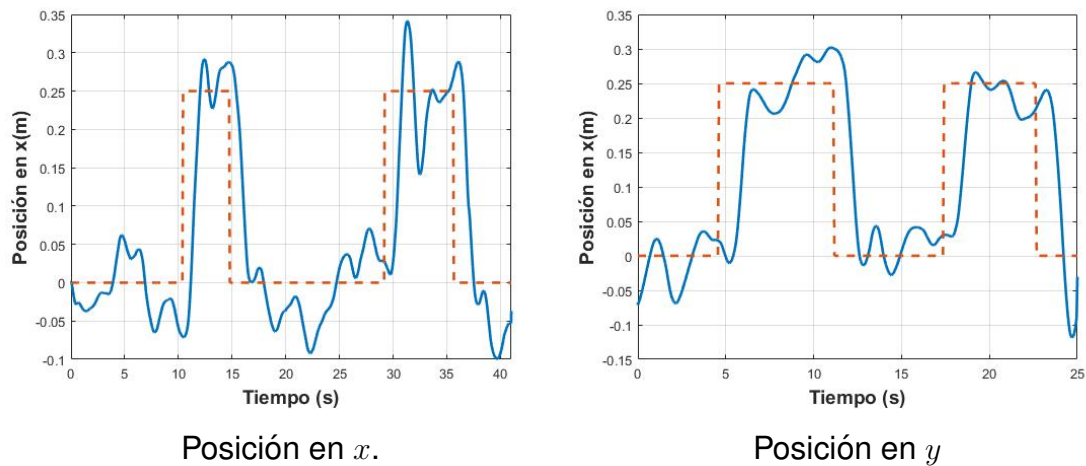


Figura 4.21: Resultados del lazo de control en cascada.

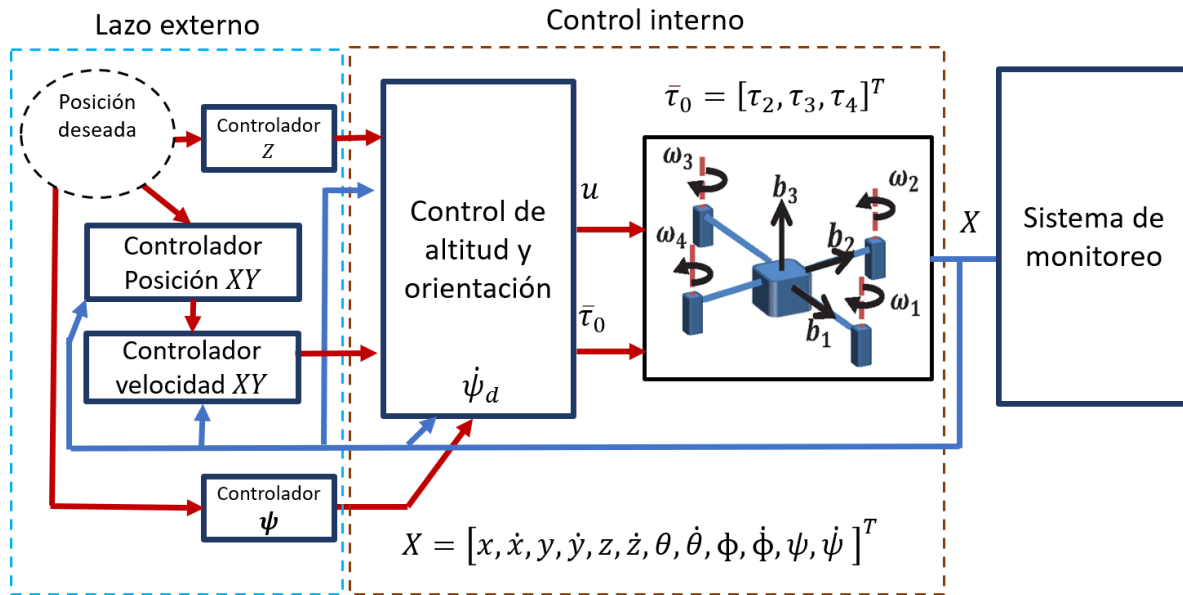


Figura 4.20: Esquema de control en cascada.

Como se muestra en la Figura 4.21  $x$  presentó un error máximo de 10 cm, pero pese a la magnitud del error, convergió nuevamente a la referencia, mientras en  $y$  el sobre impulso máximo fue de 5 cm. De los tres controladores implementados en tiempo real este fue el que presentó el mejor comportamiento.

### **4.3. Conclusión**

El sistema basado en radio frecuencia validado en el presente capítulo mostró versatilidad y practicidad, se pudo implementar tanto en Windows y Matlab como Ubuntu y ROS con relativa sencillez. Su principal inconveniente es la tasa de transmisión de los datos, pero esta se puede mejorar agregando más balizas fijas. Otro inconveniente que presentó fue la pérdida momentánea del seguimiento ante la presencia de obstáculos, se recomienda usarlo en espacios cerrados y en un área libre.

Los sistemas de SLAM visual implementados pueden tener error en una escala incorrecta en los primeros segundos del vuelo, por lo que se recomienda operar el móvil en lazo abierto de manera que permita que la paquetería ajuste su estimación. Particularmente en ORB SLAM 3, gracias a la fusión de datos inerciales con los obtenidos por visión, se obtiene una estimación muy precisa de la pose.

Respecto a los controladores, en este cuadricóptero en particular solamente se pudo experimentar con controladores clásicos ya que, gracias al control interno, su dinámica converge con relativa facilidad. Las oscilaciones presentadas en vuelo no son debidas a las estrategias de control, sino a problemas físicos en el móvil.

# CAPÍTULO 5

## CONCLUSIONES

# Capítulo 5

## Conclusiones

### 5.1. Conclusiones generales

En el presente trabajo de tesis se realizaron los desarrollos y se validaron experimentalmente estrategias de control aplicadas a un VANT, se implementaron 3 lazos de control diferentes, uno de ellos no lineal. Además, se implementaron sistemas de navegación, mapeo y localización en tiempo real. De esta forma se cumplieron todos los objetivos declarados al inicio del proyecto, con la diferencia que el calendario inicial contemplaba culminar un semestre antes y realizar una estancia, pero no se realizó así debido a las circunstancias especiales derivadas del brote de Covid-19.

Las técnicas de control diseñadas, simuladas e implementadas representaron retos significativos, los VANTs son sistemas dinámicos complejos, expuestos a incertidumbre por dinámicas no modeladas y perturbaciones de origen, tanto internas como externas. También es importante mencionar que los trabajos previos en la plataforma AR Drone o bien ya eran obsoletos o no cuentan con documentación al público.

### 5.2. Conclusiones de la experimentación

Los resultados de la experimentación sobre la plataforma AR Drone 2.0 indican que aun es posible implementar controladores sobre ella, con la premisa de que cuenta con un control interno de fábrica para ciertas variables, lo cual es una

limitante en el proceso de diseño y validación de las estrategias de control.

La plataforma como tal se encuentra obsoleta en lo referente a paqueterías y controladores para manejarla, los mas recientes datan del 2016, y ya que la mayoría son de acceso libre no se les ha dado mantenimiento ni han residido actualizaciones.

El principal inconveniente en la plataforma es la compatibilidad con los entornos actuales de desarrollo, para poder trabajarla en Windows con Matlab y Simulink se recomienda utilizar una versión entre la 2016b y la 2018b. Es posible encontrar proyectos con esta plataforma utilizando LabView, pero en esta tesis no se abordó esa ruta de acción.

Para manipularlo en Linux Ubuntu con ROS se recomienda utilizar la versión 16.06 de Ubuntu y ROS Kinetic, con el inconveniente de que esta versión de ROS caducó el presente año. El controlador del AR Drone se compila y ejecuta sin inconvenientes en esta versión, los problema surgen cuando se requiere utilizar otra paqueterías mas actuales, como fue el caso de ORB SLAM2 y ORB SLAM3, debido a la incompatibilidad entre sus compiladores. En el caso particular de utilizarlo con PTAM se compiló y ejecutó sin problemas mayores, TUM Ardrone se compiló para una versión anterior a ROS Kinetic pero en los se encuentra una versión editada por los usuarios donde se resolvieron la mayoría de los problemas de compatibilidad.

Una vez que se tiene ejecutable el controlador de AR Drone es fácil de encontrar controladores en lazo abierto basados en éste, caso como AR Drone autopilot y AR Drone tutorials, ambos escritos en Python, si se cuenta con la suficiente destreza en programación en Python y en ROS es posible editarlos para crear controladores en lazo cerrado, exactamente eso fue lo que se realizó en TUM AR Drone, se tomó AR Drone tutorias y se fusionó con el proyecto de PTAM. Dado a que el control se ejecuta en la computadora de la estación de control, si su equipo cuenta con los suficientes recursos, se posibilita el diseño de controladores más complejos con mas costo computacional.

En el caso particular del equipo utilizado en este proyecto de tesis se documentó en las entregas parciales que presenta fuertes inestabilidades ocasionales en vuelo sin razón aparente. Para el segundo semestre ya se había validado una paquetería para programar en Windows y Matlab, fue cuando se identificó esta problemática, una posible causa de este error son los grandes tiempos sin recibir comandos de control, por ese motivo se decidió explorar las

plataformas en Linux, ya que es mucho menor el costo computacional que aporta el sistema operativo, y dado a que ROS se ejecuta directo en terminal no cuenta con la carga adicional que representa las interfaces gráficas, pero el problema se presentó también en Linux. Además, se reportó que presenta oscilaciones sostenidas en los vuelos, estas oscilaciones se atenúan con juegos de hélices nuevas y con calibración de los sensores, pero no se eliminan, algo recomendado es poner al mínimo los parámetros de vuelo.

Si se compara trabajar en Windows con Matlab y en Ubuntu con ROS, es mucho mas sencillo trabajar en Windows, donde se probó correr la paquetería de Matlab para el AR Drone en Ubuntu para aprovechar las prestaciones computacionales de este último, pero la paquetería que permite la comunicación en tiempo real por WiFi no es compatible en este entorno. Es mucho mas potente trabajar en Ubuntu, pero esta potencia esta ligada uno a uno con el nivel de dificultad y con la falta de documentación, ya que en esta plataforma la mayoría de los trabajos son de código abierto.

### **5.3. Trabajos futuros**

En en presente trabajo de presentaron 2 paqueterías de SLAM visual (PTAM y ORB SLAM) y un sistema de navegación basado en radiofrecuencia, pero solamente se logró realizar el lazo cerrado con PTAM, siendo este el que presenta menor precisión. En trabajos futuros se recomienda realizar el lazo cerrado con ORB SLAM 3 simultáneamente con el sistema basado en radio frecuencia, de esa manera se tendrá un sistema de monitoreo y lazo cerrado mucho más robusto.

Además de trabajar con Parrot AR Drone, se probó la paquetería de Matlab Parrot Minidrones Support from Simulink con Parrot Mambo, con esta paquetería es posible diseñar e implementar en tiempo real controladores con distintos mini drones de la marca Parrot, con la diferencia que se cuenta con la libertad de diseñar controladores para los seis grados de libertad y el control diseñado se ejecuta dentro del sistema de control embebido, la paquetería se encuentra vigente. También en Matlab se encuentra la paquetería Ryze Tello Drone Support from MATLAB, aunque esta no se validó.

Para trabajar en ROS se encuentran disponibles las tarjetas NAVIO2 y PIXHAUK, las cuales son de código abierto, y se encuentran vigentes sus controladores. Si se



desea solamente trabajar en Linux sin ROS, La empresa Parrot publicó la versión 3 de su kit de desarrollo de software junto al entorno de programación Olympe, con el cual es posible diseñar aplicaciones de control para sus cuadrirrotores actuales, con la premisa de que también cuentan con control interno similar al del Parrot AR Drone.

Preferentemente se recomienda migrar de AR Drone a otra plataforma VANT más actualizada, para fines de implementación de controladores con bajo costo computacional se recomienda utilizar Matlab Parrot Minidrones Support from Simulink. Para desarrollo y uso de paqueterías mas potentes se recomienda trabajar en Linux con ROS con las versiones actuales de Pixhawk o Navio.

Referente a las técnicas de control en la literatura se encuentra una gran variedad de técnicas de control disponibles a implementar, buena parte de ellas esta basada en análisis de energía, por lo cual sera necesario actualizar el modelo matemático a este formato. Además, si se busca eliminar del modelo matemático las indeterminaciones se recomienda migrar a una representación de las rotaciones mediante cuaterniones.

Actualmente la investigación aplicada en VANTs sigue vigente, se exploran muchas aplicaciones como el transporte de cargas, trabajo colaborativo y aplicaciones especiales como el uso en primera respuesta a siniestros, todas estas aplicaciones pueden ser líneas que den continuidad a este trabajo de tesis.

# Referencias

- [1] Falcon P, Barreiro A, Cacho M, "Modeling of Parrot Ardrone and passivity-based reset control", IEEE EXPLORE, pp 13-19 , Spain, 2013.
- [2] P. Castillo, R Lozano, P. García y A Dzul. Robust prediction-based control for unstable delay systems. Conference on desition and control, Mauí, Hawai, 2003, pp 4014-419, Diciembre 2003.
- [3] P. Castillo, R. Lozano, A. Dzul Stabilization of a mini-rotorcraft having four rotors
- [4] P. Castillo. Control Strategy Using Vision for the Stabilization of an Experimental PVTOL Aircraft Setup. Proceedings IEEE, International conference on intelligent Robots and systems, Sendal, Japon pp 2693-2698, Septiembre 2004.
- [5] P. Castillo, P. Albertos, P. Garcia and R. Lozano Simple Real-time Attitude Stabilization of a Quad-rotor Aircraft with Bounded Signals. Proceedings of the 45th IEEE Conference on Decision Control, San Diego, CA, pp 1533-1539, 2006.
- [6] S. Bouabdallah, A. Noth and R. Siegwart, "Design and Control of an Indoor Micro Quadrotor", IEEE International Conference on Robotics and Automation (ICRA 2004) vol 5, pp 4393 - 4398, New Orleans, LA, USA, 26 April-1 May, 2004.
- [7] S. Bouabdallah, A. Noth and R. Siegwart, "PID vs LQ control techniques applied to an indoor micro quadrotor,"2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) (IEEE Cat. No.04CH37566), 2004, pp. 2451-2456 vol.3, doi: 10.1109/IROS.2004.1389776.
- [8] Bouabdallah, S., Siegwart, R. (2005, April). Backstepping and sliding-mode techniques applied to an indoor micro quadrotor. In Proceedings of the 2005 IEEE international conference on robotics and automation (pp. 2247-2252). IEEE.
- [9] S. Bouabdallah and R. Siegwart, "Full control of a quadrotor,"2007 IEEE/RSJ International Conference on Intelligent Robots and Systems, 2007, pp. 153-158, doi: 10.1109/IROS.2007.4399042.

- [10] , D. Lara, A. Sanchez, R. Lozano and P. Castillo, Real-time embedded control system for VTOL aircrafts: Application to stabilize a quad-rotor helicopter,"2006 IEEE Conference on Computer Aided Control System Design, 2006 IEEE International Conference on Control Applications, 2006 IEEE International Symposium on Intelligent Control, 2006, pp. 2553-2558, doi: 10.1109/CACSD-CCA-ISIC.2006.4777040.
- [11] , D. Lara, G. Romero, A. Sanchez and R. Lozano, "Parametric Robust Stability Analysis for Attitude Control of a Four-rotor mini-rotorcraft,"Proceedings of the 45th IEEE Conference on Decision and Control, 2006, pp. 4351-4356, doi: 10.1109/CDC.2006.377762.
- [12] , E. Rondon, I. Fantoni-Coichot, A. Sanchez and G. Sanahuja, "Optical flow-based controller for reactive and relative navigation dedicated to a four rotor rotorcraft,"2009 IEEE/RSJ International Conference on Intelligent Robots and Systems, 2009, pp. 684-689, doi: 10.1109/IROS.2009.5354483.
- [13] , García Carrillo, L.R., Rondon, E., Sanchez, A. et al. Stabilization and Trajectory Tracking of a Quad-Rotor Using Vision. *J Intell Robot Syst* 61, 103–118 (2011). <https://doi.org/10.1007/s10846-010-9472-1>
- [14] Elias Reyes-Valeria, Rogerio Enriquez-Caldera, Sergio Camacho-Lara, Jose Guichard. LQR Control for a Quadrotor using Unit Quaternions: Modeling and Simulation. *CONIELECOMP* , 23rd, pp 171-178, 2013.
- [15] Anand Sanchez, Vicente Parra-Vega, Octavio Garcia, Francisco Ruiz-Sanchez and L. E. Ramos-Velasco Time-Parametrization Control of Quadrotors with a Robust Quaternion-based Sliding Mode Controller for Aggressive Maneuvering. *European Control Conference*, Zürich, Switzerland, 2013.
- [16] Hugo Romero, Sergio Salazar · Rogelio Lozano, Omar Santos, Real-time Stabilization of a Quadrotor UAV: Nonlinear Optimal and Suboptimal Control. *J Intell Robot Syst*, pp 79-91, Mexico DF, junio 2012.
- [17] Iván González · Sergio Salazar · Rogelio Lozano, Chattering-Free Sliding Mode Altitude Control for a Quad-Rotor Aircraft: Real-Time Application. *J Intell Robot Syst*, Ed 73, pp 137-157, Mexico, DF Octubre 2013.

- [18] Aldo-Jonathan Muñoz-Vázquez, Vicente Parra-Vega, Anand Sánchez-Orta, Octavio Garcia y Carlos Izaguirre-Espinoza Attitude Tracking Control of a Quadrotor based on Absolutely Continuous Fractional Integral Sliding Modes. Conference on Control Applications, Antibes, France, pp 717-722, Octubre 2014.
- [19] Hao Liu, Xiafu Wang, and Yisheng Zhong, Quaternion-Based Robust Attitude Control for Uncertain Robotic Quadrotors, TRANSACTIONS ON INDUSTRIAL INFORMATICS, Vol 12, pp 406-416, 2015.
- [20] H. Abaunza\*, J. Cariño, P. Castillo, R. Lozano, Quadrotor Dual Quaternion Control, Workshop on Research, Education
- [21] Anand Sanchez-Orta. Vicente Parra-Vega. Carlos Izaguirre-Espinoza Position–Yaw Tracking of Quadrotors. Journal of Dynamic Systems, Measurement, and Control, Vol 137, pp 137-145, Julio 2015.
- [22] Tianqu Zhao, Hong Jiang. Landing system for AR.Drone 2.0 using onboard camera and ROS. Proceedings of 2016 IEEE Chinese Guidance, Navigation and Control Conference, Nanjing, China, pp 1098-1102, Agosto 2018.
- [23] Xuesi Li\*, Kai Jiang, Chunlei Yang, Zhihong Wang, Xuejun Zhu, Hongbo Wang, Haobin Shi, Image-based Visual Servoing for Quadrotor Helicopters using Genetic Algorithm, International Conference on Information and Automation, Wuyi Mountain, China, pp 1216- 1221 August 2018.
- [24] Tianpeng Huang, Deqing Huang, Dan Luo, Attitude tracking for a quadrotor UAV based on fuzzy PID controller, International Conference on Information, Cybernetics, and Computational Social Systems, Vol 5, pp 19-24, Chengdu, China, 2018.
- [25] Zhifang Wang , Jianguo Yu, Shangjing Lin, Junguo Dong, and zhen Yu, distributed robust adaptive fault-tolerant mechanism for quadrotor uav real-time wireless network systems with random delay and packet loss, IEE Access vol 7, Beijing, pp 134055 – 134062, China 2019
- [26] Carro I, et all, "Identification and PID Control for a Quadrocopter", Journal Telkomnika, vol.12, no.4, pp. 819-828, Puebla México, diciembre 2014.

- [27] Prayitno A., Indrawati V, Utomo G, "Trajectory Tracking of AR.Drone Quadrotor Using Fuzzy Logic Controller", Journal Telkomnika, vol.12, no.4, pp. 819-828, Diciembre 2014.
- [28] Prayitno A, et al, "Waypoint Navigation of AR.Drone Quadrotor Using Fuzzy Logic Controller", Journal Telkomnika, vol.13, no.3, pp. 930-939, Japón, Septiembre 2015.
- [29] Prayitno A, Indrawati V, Ardi-Kusuma T, "Waypoint Navigation of AR.Drone Quadrotor Using Fuzzy Logic Controller", Journal Telkomnika, vol.13, no.3, pp. 930-939, September 2015.
- [30] Santoso F, Garratt M. A., Anavatti S., G. , "Adaptive Neuro-Fuzzy Inference System Identification for the Dynamics of the AR.Drone Quadcopter", ICSEEA, ACT, Australia, 2016.
- [31] G. Martin, "Modelling and control of the parrot ar.drone. Journal of Undergraduate Engineering Research", 5(1), 2012.
- [32] Madhu-Babu V., Kaushik D, Kumar S, "Designing of Self Tuning PID Controller for AR Drone Quadrotor", IEEE Proceedings 18th, Hong Kong, China, July 2017.
- [33] M. A. Rosaldo Serrano, E. Aranda Bricaire, "Modelado y control de un prototipo comercial de aeronave tipo quadrirotor", Congreso Latinoamericano de Control Automático (CLCA 2014), pp 72-77, Quintana Roo, México, Octubre 14-17, 2014.
- [34] M. A. Rosaldo Serrano, E. Aranda Bricaire, "MODELADO Y CONTROL MEDIANTE BACKSTEPPING DE UN AR.DRONE 2.0", Memorias del XVIII Congreso Mexicano de Robotica 2016, Mazatlan Sinaloa, México, 2016.
- [35] Rosaldo-Serrano M, Santiaguillo-Salinas J, Aranda-Bricaire E, "Observer-Based Time-Varying Backstepping Control for a Quadrotor Multi-Agent System Journal of Intelligent Robotic Systems", Springer Journal of Intelligent Robotic Systems, pp 135-150, Mexico, Mexico, 2019.

- [36] Dentler J, Kannan S, Olivares M, “A real-time model predictive position control with collision avoidance for commercial low-cost quadrotors” IEEE Conference on Control Applications (CCA),pp 16-20, Buenos Aires, Argentina, 2016.
- [37] Nelson G, Muñoz P, Sellado P, Real fuzzy PID control of the UAV AR.Drone 2.0 for hovering under disturbances in known environments IEEE EXPLORE,pp 17-21, Temecu, Chile, 2017.
- [38] C Hajiyev et all, State Estimation and Control for Low-cost Unmanned Aerial Vehicles, 1a Ed, Istanbul, Turkey, Springer, 2015.
- [39] S. BOUABDALLAH, design and control of quadrotors with application to autonomous flying, DOCTORAL TESIS, FACULTAD DE CIENCIAS TECNICAS, ESCUELA POLITÉCNICA FEDERAL DE LAUSANNE, Lausanne, FRANCIA, 2007.
- [40] J. Craig, Robótica. 3ra Ed.Mexico: Prentice Hall, 2006.
- [41] K. Ogata. Ingenieria de cntrol moderna. 5ta Ed. Madrod, España. Pearson, 2010.
- [42] S Piskorski, et all. AR.Drone Developer Guide, 1 ed, Parrot compañía, Paris Francia, 2016.
- [43] Y. Shtessel, et all. Siliding Mode Control and observación.New York, 1ra Ed. USA. Spinger 2014.
- [44] P. Castillo, P. García R. Lozano, P. Albertos. Modelado y estabilización de un helicóptero con cuatro rotores, RIAI Vol. 4, pp. 41-57 Enero 2007.
- [45] Bristeau P.J., Callou, F.,Vissiere, D., Petit, N. The Navigation and Control technology inside the AR.Drone micro UAV 18th IFAC World Congress Milano, Italy, (2011)
- [46] R. Mur-Artal Y J. Tardós, ORB-SLAM2: an Open-Source SLAM System for Monocular, Stereo and RGB-D Cameras, Zaragoza España, IEEE Transactions on Robotic pp 1-5, 2015.

- [47] C. Campos, et al,ORB-SLAM3: An Accurate Open-Source Library for Visual, Visual-Inertial and Multi-Map SLAM, Zaragoza España, IEEE Transactions on Robotic pp 1-5, julio 2020.
- [48] R. Miranda, L-T Aguilar, J Herrero, Drones, modelado y control de cuadrotores, 1a. Ed, Mexico, Alfa y Omega, 2018.