

# Universidad Autónoma de Baja California

Facultad de Ingeniería, Arquitectura y Diseño  
Ensenada



---

Ingeniería en Electrónica

---

Seguimiento de trayectoria caótica esférica y formación de  
vehículos aéreos no tripulados en configuración  
maestro-esclavo

TESIS

que para cubrir parcialmente los requisitos necesarios para obtener el  
grado de

INGENIERO EN ELECTRÓNICA

presenta:

**ALEJANDRO DURÁN COVARRUBIAS**

Ensenada, Baja California, México, Enero de 2024.



Universidad Autónoma de Baja California

Facultad de Ingeniería, Arquitectura y Diseño Ensenada



Seguimiento de trayectoria caótica esférica y formación de vehículos aéreos no tripulados en configuración maestro-esclavo

**TESIS**

que para obtener el grado de ingeniero en electrónica presenta:

**ALEJANDRO DURÁN COVARRUBIAS**

Y aprobada por el siguiente comité:

Dr. César Cruz Hernández  
Director de tesis

Dr. Adrian Arellano Delgado  
Codirector de tesis

Dra. Rosa Martha López Gutiérrez  
Miembro del comité

Dra. Rigoberto Martínez Clark  
Miembro del comité

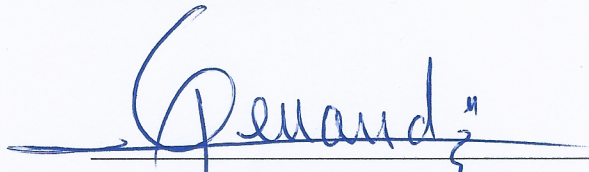
Dr. Miguel Ángel Murillo Escobar  
Miembro del comité

Ensenada, Baja California, México. Enero de 2024.

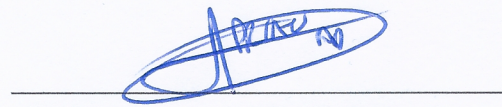
**Resumen** de la tesis de **Alejandro Duran Covarrubias**, presentada como requisito parcial para obtener el grado de INGENIERO EN ELECTRÓNICA. Ensenada B.C. México, Enero de 2024.

**Seguimiento de trayectoria caótica esférica y formación de vehículos aéreos no tripulados en configuración maestro-esclavo**

Resumen aprobado por:



Dr. César Cruz Hernández  
Director de tesis



Dr. Adrian Arellano Delgado  
Codirector de tesis

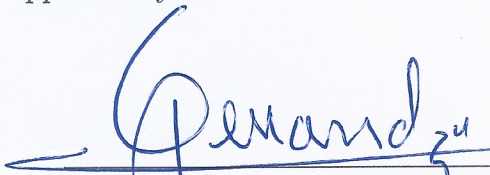
Este trabajo de tesis de licenciatura se presenta la sincronización y formación entre dos cuadricópteros en configuración maestro-esclavo con la tarea de seguimiento de una trayectoria caótica esférica. Se utiliza el método de control backstepping para el control individual de cada cuadricóptero y el acoplamiento dinámico para la sincronización y formación entre los cuadricópteros. Se utilizaron herramientas computacionales para la simulación, verificación y solución del problema propuesto. Se propusieron trayectorias en el espacio de dos y tres dimensiones, y trayectorias caóticas. Los resultados obtenidos muestran que los cuadricópteros se sincronizan y forman manteniendo estabilidad y siguen la ruta que se les indica.

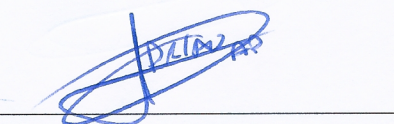
**Palabras clave:** Cuadricóptero, Control Backstepping, Acoplamiento Dinámico, Sistema Caótico Esférico.

**Abstract** of the thesis presented by **Alejandro Duran Covarrubias**, as a partial requirement to obtain the ELECTRONIC ENGINEERING Degree. Ensenada B.C. Mexico, January 2024.

**SPHERICAL CHAOTIC TRAJECTORY FOLLOWING AND  
FORMATION OF UNMANNED AERIAL VEHICLES IN  
MASTER-SLAVE CONFIGURATION**

Approved by:

  
Dr. César Cruz Hernández  
Thesis Director

  
Dr. Adrian Arellano Delgado  
Thesis Co-director

This bachelor thesis work presents the synchronization and formation between two quadcopters in master-slave configuration with the task of following a spherical chaotic trajectory. The backstepping control method is used for individual control of each quadcopter and dynamic coupling is used for synchronization and formation between the quadcopters. Computational tools were used for simulation, verification and solution of the proposed problem. Trajectories in two- and three-dimensional space and chaotic trajectories were proposed. The results obtained show that the quadcopters synchronize and form while maintaining stability and follow the path indicated to them.

**Keywords:** Quadcopter, Backstepping Control, Dynamic Coupling, Spherical Chaotic System

A mis padres

## Agradecimientos

A mi madre Alicia y a mi padre Alejandro, por ayudarme a superarme cada día más y por su amor incondicional.

Al Dr. Adrian Arellano Delgado, por su ayuda, tolerancia y retroalimentación para la realización de esta tesis.

Al Dr. César Cruz Hernández, por sus clases de Aplicaciones de Caos que me motivaron a realizar la tesis.

A la Dra. Rosa Martha López Gutiérrez, por su apoyo y guía durante todo mi trayecto universitario como tutora.

A mis amigos y compañeros de carrera por los momentos compartidos y por brindar motivación para seguir adelante.

A la Universidad Autónoma de Baja California (UABC) y a su personal, por brindar herramientas para mi formación profesional como ingeniero electrónico.

Al Centro de Investigación Científica y de Estudios Superiores de Ensenada (CI-CESE), por brindarme un apoyo económico durante mi trabajo.

Al Consejo Nacional de Humanidades, Ciencia y Tecnología (CONAHCYT) por el apoyo económico recibido a través del proyecto de Investigación en Ciencia Básica entre instituciones, Sincronización de Sistemas Complejos y Algunas Aplicaciones. Ref. 166654 y continuación (A1-S-31628).

Ensenada, B.C., México  
Enero de 2024

**Alejandro Durán Covarrubias**

# Tabla de Contenido

Resumen	iii
Abstract	iv
Dedicatoria	v
Agradecimientos	vi
Tabla de Contenido	vii
Índice de figuras	viii
<b>1. Introducción</b>	<b>1</b>
1.1. Problema de estudio . . . . .	2
1.2. Objetivo . . . . .	3
1.3. Hipótesis . . . . .	3
<b>2. Marco Teórico</b>	<b>4</b>
2.1. Sistema complejo . . . . .	4
2.2. Control Backstepping . . . . .	5
2.3. Sincronización . . . . .	6
2.4. Acoplamiento dinámico . . . . .	7
2.5. Sistemas Híbridos Dinámicos . . . . .	7
<b>3. Modelo de control</b>	<b>8</b>
3.1. Modelo del cuadricóptero . . . . .	8
3.2. Control backstepping del cuadricóptero . . . . .	13
3.2.1. Control del subsistema de rotación . . . . .	14
3.2.2. Control del subsistema de traslación . . . . .	15

3.3. Acoplamiento dinámico . . . . .	17
3.4. Cálculo Anti-colisión . . . . .	20
<b>4. Esfera Caótica</b>	<b>24</b>
4.1. Generación de la esfera . . . . .	24
4.1.1. Atractor esférico . . . . .	25
4.2. Implementación . . . . .	26
<b>5. Resultados</b>	<b>29</b>
5.1. Detalles computacionales . . . . .	29
5.2. Trayectorias en dos dimensiones . . . . .	30
5.2.1. Trayectoria Lemniscata de Bernoulli . . . . .	30
5.2.2. Trayectoria circular . . . . .	31
5.2.3. Trayectoria trisectriz caracol . . . . .	32
5.3. Trayectorias en tres dimensiones . . . . .	33
5.3.1. Trayectoria Esférica . . . . .	33
5.3.2. Trayectoria de media luna . . . . .	35
5.4. Trayectorias caóticas . . . . .	35
5.4.1. Esfera caótica . . . . .	35
5.4.2. Toro caótico . . . . .	36
<b>6. Conclusiones</b>	<b>37</b>
6.1. Trabajo Futuro . . . . .	38
<b>Bibliografía</b>	<b>39</b>
<b>A. Código para la simulación del control de los cuadricópteros y sus configuraciones</b>	<b>43</b>

# Índice de figuras

3.1.	Representación del cuadricóptero $B$ y su marco de referencia $E$ . Las flechas indican el movimiento la dirección de giro de cada rotor. . . .	9
3.2.	Interacción entre los dos subsistemas del sistema dinámico del cuadricóptero. . . . .	13
3.3.	Diagrama a bloques del controlador propuesto. . . . .	14
3.4.	Diagrama a bloques del control por backstepping. . . . .	17
3.5.	Relación de sistemas. . . . .	19
3.6.	Representación de la posición deseada entre el esclavo y maestro. . . .	22
3.7.	Sistema esclavo del cuadricóptero con el subsistema de anti-colisión incluido. . . . .	23
4.1.	Esfera caótica generada utilizando las constantes 4.4. . . . .	26
4.2.	Organización de los sistemas y sus interacciones. . . . .	28
5.1.	Trayectoria de los cuadricópteros: Lemniscata. . . . .	30
5.2.	Trayectoria de los cuadricópteros: Circular. . . . .	31
5.3.	Trayectoria de los cuadricópteros: Trisectriz. . . . .	32
5.4.	Divisiones de la esfera. . . . .	34
5.5.	Trayectoria de los cuadricópteros: Esférica. . . . .	34
5.6.	Trayectoria de los cuadricópteros: Media Luna. . . . .	35
5.7.	Trayectoria de los cuadricópteros: Esfera caótica. . . . .	36
5.8.	Trayectoria de los cuadricópteros: Toro caótico. . . . .	36



# Capítulo 1

## Introducción

Un vehículo aéreo no tripulado (UAV por sus siglas en inglés) es una aeronave sin tripulación humana que ejerce su función remotamente, conocidos también como dron. En la categoría de los drones se encuentran los cuadricópteros que se caracterizan por tener cuatro rotores y tener 6 grados de libertad. Los cuadricópteros se clasifican como helicópteros debido a que su elevación es generada por hélices orientadas verticalmente. Estos vehículos son reutilizables capaces de mantener de manera autónoma un nivel de vuelo controlado y sostenido.

El uso de vehículos aéreos no tripulados ha ido creciendo, ya que sustituye el trabajo que realizan vehículos aéreos tripulados en áreas generales y específicas como: mapeo topográfico, monitoreo de tráfico de automóviles, monitoreo del clima, rastreo y mapeo de volcanes activos, la industria militar y como soporte en diferentes áreas científicas.

En los últimos años, la investigación sobre uso de un solo vehículo aéreo no tripulado y su comportamiento se encuentra extensamente estudiado, sin embargo, es de gran interés el estudio del comportamiento con varios vehículos aéreos no tripulados así como las interacciones entre ellos, donde se destaca la autonomía para realizar tareas en ambientes hostiles donde la comunicación con un controlador remoto resulte ser inestable, costosa o inexistente.

En beneficio de algunas aplicaciones, el movimiento estable del cuadricóptero es imprescindible para asegurar el control y estabilidad general. Sin embargo, el cuadricóptero es considerado un sistema subactuado e inestable. Debido a que el modelo dinámico es un sistema no lineal compuesto de dos subsistemas interdependientes

como lo es: el de ángulos y traslación; que requieren de técnicas especializadas, de esta forma se utiliza backstepping [1] para estabilizar los subsistemas y tener un comportamiento estable y sostenido.

Como se busca usar más de un cuadricóptero, se requiere definir las interacciones en grupo entre ellos, de esta forma se utiliza el acoplamiento dinámico Maestro-Eslavo [2] debido a que nos permite definir un rol a cada cuadricóptero donde el maestro define y sigue la ruta, y el esclavo sigue al maestro.

La trayectoria que plantea seguir tiene como requerimiento prevenir cualquier tipo de seguimiento, sin embargo, el control terrestre necesita predecir aproximadamente la trayectoria del cuadricóptero, de modo que, se requiere asegurar el funcionamiento y continuidad de la tarea sin la necesidad de intervención de un control externo. Para esto se utiliza un sistema complejo que bajo ciertas condiciones nos permite tener caos, de esta forma obtenemos un comportamiento aparentemente aleatorio pero a la vez determinista.

La tesis está constituida por los siguientes capítulos: el capítulo 2 presenta el marco teórico donde se abordan los temas principales en los que la tesis se apoya. El capítulo 3 plantea el modelo dinámico del cuadricóptero así como el control por backstepping, sincronización y formación. El capítulo 4 aborda las bases teóricas de la ruta caótica esférica que seguirán los cuadricópteros. El capítulo 5 se muestran los resultados numéricos de las simulaciones de manera gráfica. Y el capítulo 6 se muestran las conclusiones finales del trabajo así como el trabajo a futuro.

## 1.1. Problema de estudio

La interacción y movimiento de una flota de cuadricópteros debe de ser autónoma y sus trayectorias fácilmente predecibles por un controlador externo sin que este tenga contacto directo con los vehículos aéreos no tripulados. A su vez que actores no puedan predecir la ruta del cuadricóptero.

Generalmente, las flotas de cuadricópteros son coordinadas por un controlador externo el cual define la trayectoria de cada uno, esto funciona bien en la mayoría de los casos, pero generalmente en algunas aplicaciones esta falta de autonomía puede ocasionar la perdida parcial o total de la flota así como la destrucción intencionada de los mismos por parte de actores externos ajenos al controlador.

Otorgarle autonomía a una flota completa de cuadricópteros, resultaría en una

mayor supervivencia en entornos hostiles y reducción las perdidas. Para esto un sistema de acoplamiento dinámico maestro-esclavo implementado ya sea electrónicamente o programado en el mismo firmware del cuadricóptero nos permitiría coordinar las relaciones entre la misma flota. A su vez el uso de un sistema caótico esférico para la generación de una trayectoria impediría que actores externos puedan rastrear y predecir la ruta de la flota, a excepción del controlador que puede seguir a los cuadricópteros sin necesidad de una comunicación directa.

## 1.2. Objetivo

Con la realización de la presente propuesta de tesis de licenciatura, se pretende alcanzar el siguiente objetivo general: Aplicar control por backstepping a una red de dos cuadricópteros en configuración maestro-esclavo con el fin de alcanzar el seguimiento, sincronización y formación a una trayectoria caótica esférica

## 1.3. Hipótesis

Es posible alcanzar sincronización y formación entre dos robots móviles acoplados en configuración maestro-esclavo con el fin de alcanzar el seguimiento, sincronización y formación a una trayectoria caótica esférica.

# Capítulo 2

## Marco Teórico

A continuación se exponen las bases teóricas en las cuales se fundamenta esta tesis de licenciatura.

### 2.1. Sistema complejo

En este gran mundo interconectado, muchos fenómenos exhiben comportamientos extraños, difíciles de predecir o incluso que parecen ser aleatorios además de que son complicadas de explicar de forma común. Esto va desde sistemas biológicos, redes neuronales hasta economía y civilizaciones humanas. A todos estos sistemas con comportamientos extraños se les define colectivamente como “sistemas complejos”. Estos sistemas complejos están caracterizados por tener muchos componentes que interactúan entre sí y son interdependientes. Debido a esto su descomposición en sistemas simples resulta complicado o imposible, pues la interdependencia con otros componentes les otorga características y comportamientos que son difíciles de predecir y entender.

Los sistemas complejos se desprenden de los sistemas no lineales. Un sistema no lineal son aquellos que no cumplen con la propiedad de superposición de los sistemas lineales. Por lo que el cambio en la entrada de un sistema no lineal no garantiza que haya un cambio en la salida del mismo[3]. Los sistemas no lineales pueden presentar al menos uno de los siguientes fenómenos [4]:

- Tiempo de escape finito: El estado de un sistema no lineal se aproxima al infinito en un tiempo finito.

- Múltiples equilibrios aislados: El sistema puede converger en múltiples estados estacionarios.
- Ciclo limite: El sistema puede oscilar en una amplitud y frecuencia fija independiente de sus valores iniciales.
- Harmónicas, subarmónicas y oscilaciones casi periódicas: Un sistema no lineal cuando es sometido a una entrada oscilatoria con una frecuencia fija, este tiene una salida que oscila en frecuencias múltiples de la frecuencia de entrada o en una suma de frecuencias que no se relacionan con la frecuencia de entrada.
- Caos: Un sistema no lineal puede tener un estado estacionario el cual no sea oscilación periódica, semi-periódica o equilibrio. Este comportamiento es referido como caos y generalmente exhibe comportamiento aleatorio.
- Múltiples modos de comportamiento: Un sistema no lineal puede presentar diferentes modos de comportamiento, puede exhibir caos, oscilaciones o equilibrio dependiendo de la amplitud y frecuencia en la entrada del sistema.

Un sistema es complejo cuando todos o la mayoría de sus componentes tienen la propiedad de emergencia, su comportamiento no es posible de inferir por el comportamiento de sus componentes y la información necesaria para describir el sistema mide la complejidad del mismo.

La propiedad de emergencia es cuando el comportamiento y propiedades de los componentes de un sistema no describen el comportamiento del mismo. Por lo que reducir o simplificar un sistema no es posible, así mismo aislar y estudiar los componentes por separado provoca que se comporten diferente a como lo harían dentro del sistema. Debido a esto se requiere el estudio de los componentes dentro del contexto del sistema complejo. [5]

## 2.2. Control Backstepping

En sistemas de control existen diferentes métodos de control que se han desarrollado para sistemas dinámicos lineales y no lineales los cuales se diferencian por los objetivos de diseño, en el campo de sistemas dinámicos no lineales existe el control backstepping.

El control backstepping es un procedimiento de diseño recursivo que conecta la selección la función de Lyapunov de control con el diseño de retroalimentación del

sistema el cual garantiza una estabilidad asintótica. Este tipo de controlador generalmente no tiene las limitaciones de linealización de un sistema no lineal, debido a que este proceso generalmente no toma en cuenta cuando el sistema exhibe múltiples modos de comportamiento o caos.

La metodología para aplicar el control backstepping en un sistema no lineal es el siguiente:

1. Se busca descomponer el sistema no lineal general en subsistemas más pequeños y coherentes.
2. Se trabaja estabilizando los subsistemas, empezando desde el subsistema relacionado con la salida.
3. Después de diseñar la ley de control para el primer subsistema, obtendremos un nuevo término de error el cual será alimentado al control del siguiente subsistema.
4. Se desarrollan los controles virtuales de los demás subsistemas.
5. Se combinan cada una de las leyes de control desarrollado para cada subsistema, dando como resultado la estabilización completa del sistema.

El uso del control backstepping es útil al momento de implementar diseños de control en sistemas con múltiples variables de estado debido a que puede manejar sistemas complejos y dinámicas erráticas o inciertas. [1]

## 2.3. Sincronización

La sincronización es un fenómeno de los sistemas que ocurre cuando estos ajustan su comportamiento para que ocurran en armonía o al unísono. Este fenómeno depende de las relaciones de señales y parámetros de los sistemas para obtener una sincronización exitosa. Se dice que uno o varios sistemas se encuentran fuera de sincronización cuando la salida del sistema no coincide con el sistema sincronizado. Este fenómeno a pesar de ser una característica natural de muchos sistemas, existen diferentes métodos para inducir sincronización de diferentes sistemas y estos aplican desde sistemas lineales a no lineales. [6]

Un ejemplo de sincronización es el experimento de Huygens, donde él observó que si dos relojes de péndulos que se encuentran sujetos en una misma base en

común inician con diferente fuerza o en diferente tiempo, sus oscilaciones llegarán a un punto donde los dos se sincronizan.

## 2.4. Acoplamiento dinámico

Existen diferentes métodos para inducir sincronización en un sistema que naturalmente no lo tiene, como es el caso de los sistemas no lineales y caóticos.

En la literatura de control se usa la dinámica maestro-esclavo para referirnos a un control unidireccional donde se utilizan métodos de acoplamiento donde el sistema maestro envía una señal para retroalimentar al sistema esclavo.

Uno de los métodos es el acoplamiento estático [7], el cual nos permite tener un acoplamiento utilizando una constante que multiplica la diferencia de las señales de dos estados como error en el sistema esclavo. La constante es conocida como fuerza de acoplamiento.

Por otro lado, el acoplamiento dinámico es una forma de sincronización entre dos sistemas no lineales iguales, tiene la flexibilidad de que no se requiere saber en su totalidad el sistema a controlar. Esta forma de acoplamiento está basada en la retroalimentación de salida de teoría de control [8], el cual nos permite trabajar sobre el sistema a sincronizar sin modificar los subsistemas internos únicamente agregando un controlador externo. El controlador externo consiste en un sistema que calcula de manera dinámica la señal de retroalimentación. [9]

## 2.5. Sistemas Híbridos Dinámicos

Los sistemas híbridos dinámicos exhiben comportamiento de sistemas dinámicos de tiempo continuo generalmente descritos por ecuaciones diferenciales y sistemas dinámicos de eventos discretos descrito generalmente por condicionales. Estos dos tipos de sistemas interactúan el uno con el otro y generan comportamiento dinámico complejo. Este tipo de sistemas aparecen de manera natural al modelar sistemas de control avanzados, puesto que se combinan múltiples controles simples en un solo control complejo para obtener un mejor rendimiento, respuesta al error o manejar diferentes escenarios de operación.

Generalmente, los subsistemas continuos y discretos se analizan por separado, sin embargo, investigar y analizar los subsistemas en conjunto es importante para encontrar la estabilización del sistema, puesto que un subsistema discreto puede llevar a la desestabilización de sistemas continuos estables. [10]

# Capítulo 3

## Modelo de control

En este capítulo se expone la teoría necesaria para obtener los resultados de la tesis.

### 3.1. Modelo del cuadricóptero

Para realizar un análisis, es necesario desarrollar un modelado matemático que aproxime el comportamiento físico de lo que se quiera estudiar, todo esto con la intención de facilitar el desarrollo de prototipos físicos funcionales, además, herramientas como la simulación permiten observar el comportamiento en la vida real sin necesidad de tener el objeto físico a la mano. El modelo matemático de sistemas y objetos reales facilita el desarrollo en áreas de robótica, electrónica, mecánica, fluidos, etc.

Un cuadricóptero es un vehículo volador de despegue y aterrizaje vertical, el cual cuenta con 4 motores para propulsarse hacia arriba, proporcionándole más velocidad que los vehículos aéreos propulsados de un solo motor. El modelo del cuadricóptero, así como su desarrollo se basa en [11].

Los cuadricópteros son sistemas subactuados, provocando que tengan comportamiento no lineal y sean susceptibles al ruido. El cuadricóptero no se diferencia mucho de otros vehículos aéreos, puesto que tiene 6 grados de libertad, por lo que, se tendrán 6 salidas que se deben de controlar.

Una manera de ver el cuadricóptero en el espacio es como se ve en la figura 3.1 donde se considera un marco fijo  $E$  y un cuerpo fijo  $B$ . Usando los ángulos de Euler se puede determinar que la posición del cuerpo volador  $B$  es debido a la rotación  $R$

de  $B$  hacia  $E$ . Utilizando el formalismo de Newton-Euler se puede escribir el modelo dinámico del cuerpo rígido bajo las fuerzas externas aplicadas al centro de masa como se describe en la fórmula 3.1.

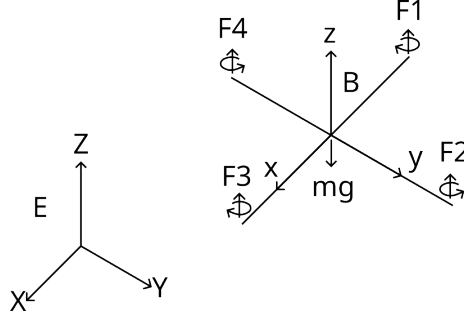


Figura 3.1: Representación del cuadricóptero  $B$  y su marco de referencia  $E$ . Las flechas indican el movimiento la dirección de giro de cada rotor.

$$\begin{bmatrix} mI_{3 \times 3} & 0 \\ 0 & I \end{bmatrix} \begin{bmatrix} \dot{V} \\ \dot{\omega} \end{bmatrix} + \begin{bmatrix} \omega \times mV \\ \omega \times I\omega \end{bmatrix} = \begin{bmatrix} F \\ \tau \end{bmatrix} \quad (3.1)$$

Donde  $I \in \mathbb{R}^{3 \times 3}$  es la matriz de inercia,  $V$  es el vector de la velocidad lineal del cuadricóptero y  $\omega$  es la velocidad angular del cuerpo. Las ecuaciones de movimiento del helicóptero se escriben como:

$$\begin{aligned} \dot{\zeta} &= v \\ m\dot{v} &= RF_b \\ \dot{R} &= R\hat{\omega} \\ J\dot{\omega} &= -\omega \times J\omega + \tau_a \end{aligned} \quad (3.2)$$

Por lo que el modelo aproximado para el cuadricóptero se puede reescribir como:

$$\begin{aligned} \dot{\zeta} &= v \\ \dot{v} &= -ge_3 + R_{e_3} \left( \frac{b}{m} \sum \Omega_i^2 \right) \\ \dot{R} &= R\hat{\omega} \\ I\dot{\omega} &= -\omega \times I\omega + \sum J_r(\omega \times e_3)\Omega_i + \tau_a \end{aligned} \quad (3.3)$$

Donde:

Simbolo	Definición
$\zeta$	Vector de posición
$R$	Matriz de rotación
$\hat{\omega}$	Matriz del segmento simétrico
$\phi$	Angulo balanceo
$\theta$	Angulo de cabeceo
$\psi$	Angulo de guiñada
$\Omega$	Velocidad del rotor
$I_{x,y,z}$	Inercia del cuerpo
$J_r$	Inercia del rotor
$J_m$	Inercia del motor
$J_p$	Inercia de la propela
$\tau_a$	Torque
$b$	Factor de Empuje
$d$	Factor de arrastre
$l$	Palanca

El torque aplicado sobre el cuerpo del vehículo sobre el eje es la diferencia entre cada torque generado por cada motor en el otro eje.

$$\tau_a = \begin{pmatrix} lb(\Omega_4^2 - \Omega_2^2) \\ lb(\Omega_3^2 - \Omega_1^2) \\ d(\Omega_4^2 + \Omega_2^2 - \Omega_3^2 - \Omega_1^2) \end{pmatrix} \quad (3.4)$$

Si se considera que la inercia del motor y que la caja de cambios inversa proporcionan una inercia imperceptible, entonces la inercia del rotor es:

$$J_r = J_p - J_m r \quad (3.5)$$

Debido al desarrollo anterior, se obtiene el modelo dinámico completo del cuadricóptero donde los movimientos en  $x$ ,  $y$  y  $z$  son consecuencia del balanceo, cabeceo y guiñada:

$$\begin{aligned}
\ddot{\phi} &= \dot{\theta}\dot{\psi} \left( \frac{I_y - I_z}{I_x} \right) - \frac{J_r}{I_x} \dot{\theta}\Omega + \frac{l}{I_x} U_2 \\
\ddot{\theta} &= \dot{\phi}\dot{\psi} \left( \frac{I_z - I_x}{I_y} \right) + \frac{J_r}{I_y} \dot{\phi}\Omega + \frac{l}{I_y} U_3 \\
\ddot{\psi} &= \dot{\phi}\dot{\theta} \left( \frac{I_x - I_y}{I_z} \right) + \frac{1}{I_z} U_4 \\
\ddot{z} &= -g + (\cos \phi \cos \theta) \frac{1}{m} U_1 \\
\ddot{x} &= (\cos \phi \sin \theta \cos \psi + \sin \phi \sin \psi) \frac{1}{m} U_1 \\
\ddot{y} &= (\cos \phi \sin \theta \sin \psi - \sin \phi \cos \psi) \frac{1}{m} U_1
\end{aligned} \tag{3.6}$$

Donde los primeros tres términos corresponden al subsistema de ángulos  $(\phi, \theta, \psi)$ , y los últimos tres términos corresponden al subsistema de traslación. Las entradas al sistema son  $U_1, U_2, U_3$  y  $U_4$  y  $\Omega$ , donde  $\Omega$  es una perturbación, por lo tanto, se pueden obtener las entradas al sistema como:

$$\begin{aligned}
U_1 &= b(\Omega_1^2 + \Omega_2^2 + \Omega_3^2 + \Omega_4^2) \\
U_2 &= b(\Omega_4^2 - \Omega_2^2) \\
U_3 &= b(\Omega_3^2 - \Omega_1^2) \\
U_4 &= d(\Omega_2^2 + \Omega_4^2 - \Omega_1^2 - \Omega_3^2) \\
\Omega &= \Omega_2 + \Omega_4 - \Omega_1 - \Omega_3
\end{aligned} \tag{3.7}$$

Para trabajar con el modelo dinámico del cuadricóptero, es necesario reescribirlo en su forma de espacio de estados, debido a que es un sistema de segundo orden este se debe convertir a uno de primer orden obteniendo así 12 variables de estado. Se introducen  $X = (x_1, \dots, x_{12}) \in \mathbb{R}^{12}$  como las variables de estado.

$$\begin{aligned}
x_1 &= \phi & x_7 &= z \\
x_2 &= \dot{x}_1 = \dot{\phi} & x_8 &= \dot{x}_7 = \dot{z} \\
x_3 &= \theta & x_9 &= x \\
x_4 &= \dot{x}_3 = \dot{\theta} & x_{10} &= \dot{x}_9 = \dot{x} \\
x_5 &= \psi & x_{11} &= y \\
x_6 &= \dot{x}_5 = \dot{\psi} & x_{12} &= \dot{x}_{11} = \dot{y}
\end{aligned} \tag{3.8}$$

Por lo que se reescribe la ecuación 3.6 de la forma  $\dot{X} = f(X, U)$  obteniendo:

$$f(X, U) = \begin{pmatrix} x_2 \\ x_4x_6a_1 + x_4a_2\Omega + b_1U_2 \\ x_4 \\ x_2x_6a_3 + x_2a_4\Omega + b_2U_3 \\ x_6 \\ x_4x_2a_5 + b_3U_4 \\ x_8 \\ -g + (\cos x_1 \cos x_3) \frac{1}{m}U_1 \\ x_{10} \\ u_x \frac{1}{m}U_1 \\ x_{12} \\ u_y \frac{1}{m}U_1 \end{pmatrix} \quad (3.9)$$

Donde:

$$\begin{aligned} a_1 &= \frac{I_y - I_z}{I_x} \\ a_2 &= \frac{-J_R}{I_x} \\ a_3 &= \frac{I_z - I_x}{I_y} \\ a_4 &= \frac{J_R}{I_y} \\ a_5 &= \frac{I_x - I_y}{I_z} \\ b_1 &= \frac{l}{I_x} \\ b_2 &= \frac{l}{I_y} \\ b_3 &= \frac{l}{I_z} \end{aligned} \quad (3.10)$$

$$\begin{aligned} u_x &= (\cos x_1 \sin x_3 \cos x_5 + \sin x_1 \sin x_5) \\ u_y &= (\cos x_1 \sin x_3 \sin x_5 - \sin x_1 \cos x_5) \end{aligned} \quad (3.11)$$

Sintetizando lo anterior, el modelo dinámico del cuadricóptero anteriormente descrito, nos permite ver que para mover el cuadricóptero se ocupan ingresar los valores

de rotación de cada motor, provocando que para movernos en el espacio resulte complicado sin usar la prueba y error. A continuación se implementará el control por backstepping para convertir valores deseados de  $\phi$ ,  $\theta$ ,  $\psi$ ,  $x$ ,  $y$  y  $z$  a las entradas del cuadricóptero.

## 3.2. Control backstepping del cuadricóptero

En relación con el sistema dinámico del cuadricóptero descrito anteriormente, se busca aplicar valores deseados de traslación y rotación, esto con la finalidad de que el cuadricóptero siga una ruta específica. Hay que tener en cuenta que el sistema del cuadricóptero es un sistema no lineal y de múltiples variables, debido a esto es necesario aplicar el método de control backstepping, obteniendo así un sistema de control sin la necesidad de modificar el sistema modelado anteriormente.

El modelo de control backstepping, así como su descripción se basa en [11].

Para realizar el control por backstepping es primordial identificar las entradas, salidas y posibles subsistemas, por consiguiente, analizando el modelo dinámico del cuadricóptero 3.9, se obtienen los subsistemas de ángulos y traslación, donde las entradas  $U_2$ ,  $U_3$ ,  $U_4$  controlan directamente el subsistema de rotación y la entrada  $U_1$  controla directamente al sistema de traslación.

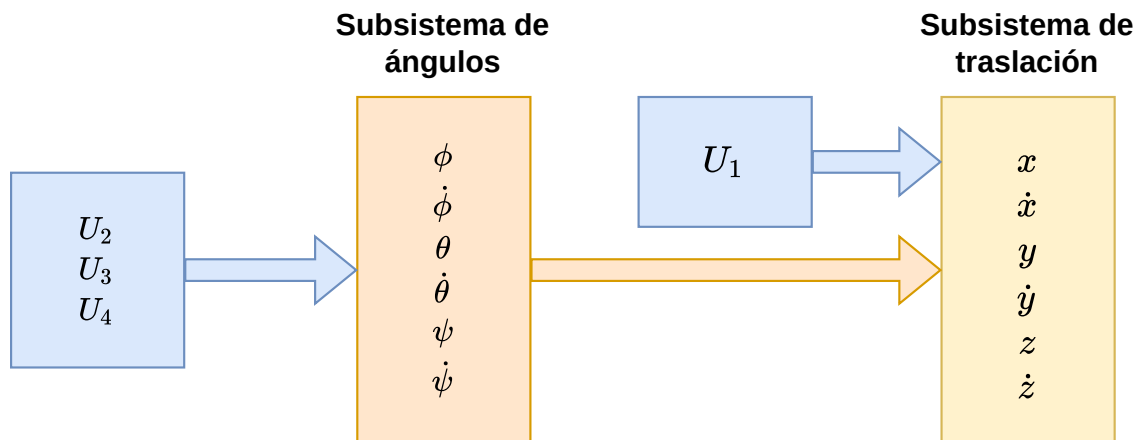


Figura 3.2: Interacción entre los dos subsistemas del sistema dinámico del cuadricóptero.

En la figura 3.2, se puede observar que el subsistema de ángulos es independiente,

en cambio, el subsistema de traslación depende del subsistema de ángulos. Para controlar los dos subsistemas, se propone el siguiente control que se divide en dos controladores: el control de posición y el control de rotación. Donde se busca que el control de posición mediante los valores deseados  $(x_d, y_d, z_d)$  genere los valores requeridos para el control de rotación y que este a su vez genere los valores para la entrada del sistema dinámico del cuadricóptero  $(U_1, U_2, U_3, U_4)$  como se puede ver en la figura 3.3.

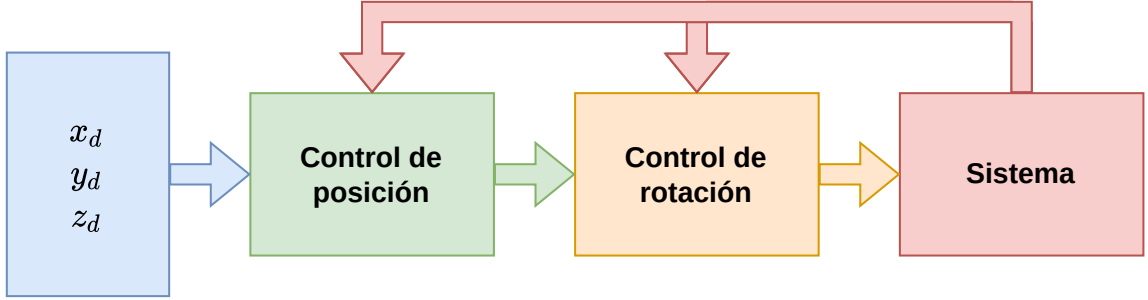


Figura 3.3: Diagrama a bloques del controlador propuesto.

### 3.2.1. Control del subsistema de rotación

Se realiza el control backstepping sobre el subsistema de rotación; donde se puede obtener la ley de control considerando que se fuerza el sistema a una trayectoria deseada. Primero se considera el error de seguimiento:

$$z_1 = x_{1d} - x_1 \quad (3.12)$$

Se utiliza el teorema de Lyapunov considerando  $V$  sea positiva y su derivada negativa:

$$V(z_1) = \frac{1}{2} z_1^2 \quad (3.13)$$

$$\dot{V}(z_1) = z_1(\dot{x}_{1d} - \dot{x}_2) \quad (3.14)$$

La estabilización de  $z_1$  se puede alcanzar si se introduce un control virtual  $x_2$ :

$$x_2 = \dot{x}_{1d} + \alpha_1 z_1 \quad \text{con: } \alpha_1 > 0 \quad (3.15)$$

Por lo que se obtiene con base en la ecuación 3.14 el control virtual de  $z_1$ :

$$z_2 = x_2 - \dot{x}_{1d} - \alpha_1 z_1 \quad (3.16)$$

Para el siguiente paso, se considera la siguiente función aumentada de Lyapunov:

$$V(z_1, z_2) = \frac{1}{2}(z_1^2 + z_2^2) \quad (3.17)$$

De la cual la derivada con respecto al tiempo es:

$$\begin{aligned} \dot{V}(z_1 z_2) &= z_2(\alpha_1 x_4 x_6 + \alpha_2 x_4 \Omega + b_1 U_2) \\ &\quad - z_2(\ddot{x}_{1d} - \alpha_1(z_2 + \alpha_1 z_1)) - z_1 z_2 - \alpha_1 z_1^2 \end{aligned} \quad (3.18)$$

La entrada de control  $U_2$  es extraída satisfaciendo  $\dot{V}(z_1 z_2) < 0$ :

$$U_2 = \frac{1}{b_1}(z_1 - a_1 x_4 x_6 - a_2 x_4 \Omega - \alpha_1(z_2 + \alpha_1 z_1) - \alpha_2 z_2) \quad (3.19)$$

De la misma forma, se aplican los pasos anteriores para obtener  $z_3, z_4, z_5, z_6$ :

$$\begin{aligned} z_3 &= x_{3d} - x_3 \\ z_4 &= x_4 - \dot{x}_{3d} - \alpha_3 z_3 \\ z_5 &= x_{5d} - x_5 \\ z_6 &= x_6 - \dot{x}_{5d} - \alpha_5 z_5 \end{aligned} \quad (3.20)$$

Obteniendo así las entradas  $U_3$  y  $U_4$ :

$$\begin{aligned} U_3 &= \frac{1}{b_2}(z_3 - a_3 x_2 x_6 - a_4 x_2 \Omega - \alpha_3(z_4 + \alpha_3 z_3) - \alpha_4 z_4) \\ U_4 &= \frac{1}{b_3}(z_5 - a_5 x_2 x_4 - \alpha_5(z_6 + \alpha_5 z_5) - \alpha_6 z_6) \end{aligned} \quad (3.21)$$

### 3.2.2. Control del subsistema de traslación

Para el movimiento de traslación se puede observar que  $U_1$  es usado para el movimiento lineal en los ejes  $XYZ$ , por lo tanto, se separa el control de altitud con el del movimiento lineal en  $X$  y  $Y$  de tal forma que realizando los pasos anteriores se obtiene:

$$\begin{aligned} z_7 &= x_{7d} - x_7 \\ z_8 &= x_8 - \dot{x}_{7d} - \alpha_7 z_7 \end{aligned} \quad (3.22)$$

Por lo tanto, se obtiene el control para  $U_1$ :

$$U_1 = \frac{m}{\cos x_1 \cos x_3}(z_7 + g - \alpha_7(z_8 + \alpha_7 z_7) - \alpha_8 z_8) \quad (3.23)$$

Para el movimiento en  $X$  y  $Y$ , se consideran los componentes 3.11 como entradas independientes al sistema, de esta forma se obtiene:

$$\begin{aligned}
 z_9 &= x_{9d} - x_9 \\
 z_{10} &= x_{10} - \dot{x}_{9d} - \alpha_9 z_9 \\
 z_{11} &= x_{11d} - x_{11} \\
 z_{12} &= x_{12} - \dot{x}_{11d} - \alpha_{11} z_{11}
 \end{aligned} \tag{3.24}$$

En consecuencia se obtiene un nuevo control para  $u_x$  y  $u_y$ :

$$\begin{aligned}
 u_x &= \frac{m}{U_1} (z_9 - \alpha_9 (z_{10} + \alpha_9 z_9) - \alpha_{10} z_{10}) \\
 u_y &= \frac{m}{U_1} (z_{11} - \alpha_{11} (z_{12} + \alpha_{11} z_{11}) - \alpha_{12} z_{12})
 \end{aligned} \tag{3.25}$$

Finalmente, los errores y sus respectivos controles obtenidos anteriormente 3.12, 3.16, 3.20, 3.22 y 3.24 se pueden representar de la siguiente forma:

$$\begin{aligned}
 z_1 &= x_{1d} - x_1 \\
 z_2 &= x_2 - x_{2d} - \alpha_1 z_1 \\
 z_3 &= x_{3d} - x_3 \\
 z_4 &= x_4 - x_{4d} - \alpha_3 z_3 \\
 z_5 &= x_{5d} - x_5 \\
 z_6 &= x_6 - x_{6d} - \alpha_5 z_5 \\
 z_7 &= x_{7d} - x_7 \\
 z_8 &= x_8 - x_{8d} - \alpha_7 z_7 \\
 z_9 &= x_{9d} - x_9 \\
 z_{10} &= x_{10} - x_{10d} - \alpha_9 z_9 \\
 z_{11} &= x_{11d} - x_{11} \\
 z_{12} &= x_{12} - x_{12d} - \alpha_{11} z_{11}
 \end{aligned} \tag{3.26}$$

Donde se considera que los valores deseados toman la misma forma que las variables de estado 3.8.

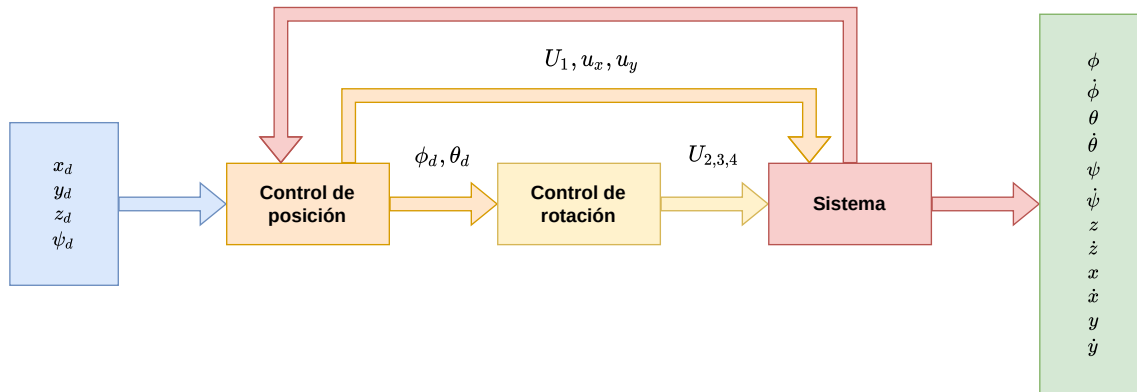


Figura 3.4: Diagrama a bloques del control por backstepping.

Igualmente, se obtiene el modelado final del sistema de control, el cual se diferencia de la lógica de control propuesta 3.3. Donde el modelado del control backstepping nos permite tener un control para el subsistema de rotación y traslación; asimismo, se pueden seleccionar las variables de  $x$ ,  $y$ ,  $z$  y  $\psi$  como valores deseados. Véase 3.4.

En resumidas cuentas el control backstepping es necesario para mantener el cuadricóptero en un vuelo estable, así como seguir una trayectoria necesaria. Este control tanto en simulación como en la vida real ha demostrado ser muy estable además de tener mucha tolerancia a las perturbaciones externas, lo cual nos garantiza que cumple con los requerimientos para moverse de forma precisa en el espacio 3D.

### 3.3. Acoplamiento dinámico

Ahora que ya podemos mover y estabilizar el cuadricóptero de acuerdo a las variables deseadas, surge la necesidad de poder controlar más de un solo cuadricóptero. Existen diversas configuraciones para poder manejar más de dos cuadricópteros al mismo tiempo, sin embargo, se busca la configuración clásica de maestro-esclavo. Esto debido a que se cumple la necesidad de controlar solamente un cuadricóptero, en este caso, el maestro, mientras que el esclavo sigue al maestro; de esta forma se logra una sincronización entre ambos cuadricópteros.

En primera instancia, se considera el cuadricóptero maestro, el cual es controlado para seguir una trayectoria deseada, posteriormente el maestro enviara su posición al esclavo, donde se plantea que la señal de acoplamiento sea directamente alimentada al error del controlador. Por otro lado, se plantea que el maestro también se acople

a los valores deseados de la ruta que se genere.

Se implementa un acoplamiento dinámico basado en un sistema de masa-resorte-amortiguador de segundo orden, donde el error generado por este es usado para acoplar los dos sistemas. [2]

$$\begin{aligned}\dot{x}_m &= F(x_m) \\ \dot{x}_s &= F(x_s) - B_1 h \\ \dot{h} &= Gh - kB_2(x_m - x_s)\end{aligned}\tag{3.27}$$

Donde  $x_m$  y  $x_s$  son el sistema maestro y esclavo respectivamente,  $\dot{h} = (h_1, h_2)^T$  son las variables de estado del sistema dinámico,  $F$  es el vector de campo que puede ser lineal o no lineal,  $B_1$  y  $B_2$  son las matrices de acoplamiento,  $k$  es la fuerza de acoplamiento y  $G$  es la matriz de acoplamiento.

Basándonos en el sistema 3.27 se obtiene que el acoplamiento dinámico es el siguiente:

$$\begin{aligned}\dot{h}_1 &= -\alpha h_1 + h_2 \\ \dot{h}_2 &= -\gamma_1 h_1 - \gamma_2 h_2 - k(x_{1m} - x_{1s})\end{aligned}\tag{3.28}$$

Donde  $h_2$  es la señal de error que alimentara al sistema de control,  $\gamma_1$  y  $\gamma_2$  son constantes positivas. Como se buscan acoplar tres variables ( $x$ ,  $y$ ,  $z$ ), se proponen los siguientes tres sistemas dinámicos:

$$\begin{aligned}\dot{h}_{1x} &= -\alpha h_{1x} + h_{2x} \\ \dot{h}_{2x} &= -\gamma_1 h_{1x} - \gamma_2 h_{2x} - k(x_m - x_s)\end{aligned}\tag{3.29}$$

$$\begin{aligned}\dot{h}_{1y} &= -\alpha h_{1y} + h_{2y} \\ \dot{h}_{2y} &= -\gamma_1 h_{1z} - \gamma_2 h_{2x} - k(y_m - y_s)\end{aligned}\tag{3.30}$$

$$\begin{aligned}\dot{h}_{1z} &= -\alpha h_{1z} + h_{2z} \\ \dot{h}_{2z} &= -\gamma_1 h_{1z} - \gamma_2 h_{2z} - k(z_m - z_s)\end{aligned}\tag{3.31}$$

Por lo tanto, se aplica la señal de acoplamiento al error del sistema de control:

$$\begin{aligned}
 z_1 &= x_{1d} - x_1 \\
 z_2 &= x_2 - x_{2d} - \alpha_1 z_1 \\
 z_3 &= x_{3d} - x_3 \\
 z_4 &= x_4 - x_{4d} - \alpha_3 z_3 \\
 z_5 &= x_{5d} - x_5 \\
 z_6 &= x_6 - x_{6d} - \alpha_5 z_5 \\
 z_7 &= -h_{2z} \\
 z_8 &= x_8 - x_{8d} - \alpha_7 z_7 \\
 z_9 &= -h_{2x} \\
 z_{10} &= x_{10} - x_{10d} - \alpha_9 z_9 \\
 z_{11} &= -h_{2y} \\
 z_{12} &= x_{12} - x_{12d} - \alpha_{11} z_{11}
 \end{aligned} \tag{3.32}$$

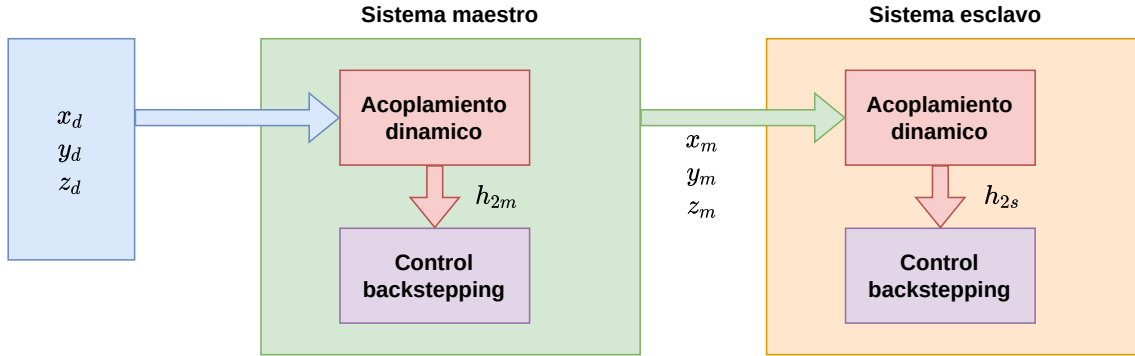


Figura 3.5: Relación de sistemas.

Consecutivamente, se presenta el acoplamiento como un subsistema más dentro del cuadricóptero, por lo tanto, la relación entre los sistemas y subsistemas se puede ver en la figura 3.5. Lo cual cumple con lo especificado al inicio de la sección. De esta forma el subsistema de acoplamiento del sistema maestro sería el siguiente:

$$\begin{aligned}
 \dot{h}_{1x} &= -\alpha h_{1x} + h_{2x} \\
 \dot{h}_{2x} &= -\gamma_1 h_{1x} - \gamma_2 h_{2x} - k(x_d - x_{m9})
 \end{aligned} \tag{3.33}$$

$$\begin{aligned}
 \dot{h}_{1y} &= -\alpha h_{1y} + h_{2y} \\
 \dot{h}_{2y} &= -\gamma_1 h_{1y} - \gamma_2 h_{2y} - k(y_d - x_{m11})
 \end{aligned} \tag{3.34}$$

$$\begin{aligned}\dot{h}_{1z} &= -\alpha h_{1z} + h_{2z} \\ \dot{h}_{2z} &= -\gamma_1 h_{1z} - \gamma_2 h_{2z} - k(z_d - x_{m7})\end{aligned}\quad (3.35)$$

Y para el sistema esclavo lo siguiente:

$$\begin{aligned}\dot{h}_{1x} &= -\alpha h_{1x} + h_{2x} \\ \dot{h}_{2x} &= -\gamma_1 h_{1x} - \gamma_2 h_{2x} - k(x_{m9} - x_{s9})\end{aligned}\quad (3.36)$$

$$\begin{aligned}\dot{h}_{1y} &= -\alpha h_{1y} + h_{2y} \\ \dot{h}_{2y} &= -\gamma_1 h_{1y} - \gamma_2 h_{2y} - k(x_{m11} - x_{s11})\end{aligned}\quad (3.37)$$

$$\begin{aligned}\dot{h}_{1z} &= -\alpha h_{1z} + h_{2z} \\ \dot{h}_{2z} &= -\gamma_1 h_{1z} - \gamma_2 h_{2z} - k(x_{m7} - x_{s7})\end{aligned}\quad (3.38)$$

En resumen el acoplamiento dinámico entre los dos cuadricópteros nos permite lograr una sincronización entre los dos sistemas con una modificación mínima al control backstepping. Este tipo de acoplamiento es el ideal para los cuadricópteros debido a que se hace uso de un subsistema extra sin agregar componentes físicos extras al cuadricóptero.

### 3.4. Cálculo Anti-colisión

Durante la etapa de simulación, se observó que el cuadricóptero maestro y esclavo se acercaban o simplemente sus trayectorias se cruzaban el uno con el otro, por lo que se vio la necesidad de un cálculo anti-colisión para evitar problemas de colisión. Además de que nos ayuda a que los cuadricópteros se mantengan a una distancia deseada.

Para calcular la distancia entre los cuadricópteros maestro y esclavo se utilizó la ecuación euclidiana de distancia de dos puntos en tres dimensiones.

$$|x_{mxyz} \ x_{sxyz}| = \sqrt{(x_{mx} - x_{sx})^2 + (x_{my} - x_{sy})^2 + (x_{mz} - x_{sz})^2} \quad (3.39)$$

Existen diferentes maneras para que los cuadricópteros mantengan distancia entre sí, uno de ellos es con el uso de un control de histéresis ON/OFF en el esclavo, detección usando sensores de proximidad como LIDAR o ultrasonico entre los cuadricópteros, control predictivo [12] e inteligencia artificial basado en modelos biológicos [13]. Sin embargo, dado que el cuadricóptero maestro comunica constantemente su posición para alimentar al sistema de acoplamiento dinámico del esclavo, se pueden

usar esas mismas coordenadas para calcular de manera directa la siguiente posición del esclavo.

Basándonos en la interpolación lineal de Newton 3.40 para calcular el punto intermedio entre dos puntos conocidos, este se puede usar para calcular de manera activa la posición del esclavo de tal forma que siempre mantendrá una distancia constante con el maestro.

$$f(x) = \frac{f(x_1) - f(x_0)}{x_1 - x_0}(x - x_0) + f(x_0) \quad (3.40)$$

La ecuación de la interpolación lineal solo es aplicable en el espacio bidimensional. Para poder trabajar con la función en el espacio tridimensional, es necesario simplificarla para que trabaje con distancias en vez de puntos.

$$f(x) = f(x_0) + \frac{d_y}{d_x}(x - x_0) \quad (3.41)$$

Posteriormente, se ajusta para que trabaje en el espacio tridimensional.

$$f(x, y, z) = f(x_0, y_0, z_0) + \frac{d_{xyz}}{d(f(x_1, y_1, z_1) - f(x_0, y_0, z_0))}(f(x_1, y_1, z_1) - f(x_0, y_0, z_0)) \quad (3.42)$$

En la ecuación 3.42,  $f(xyz)$  es el punto desconocido en el espacio 3D,  $f(x_1y_1z_1)$ ,  $f(x_0y_0z_0)$  son los puntos conocidos y  $d_{xyz}$  es la distancia de  $f(x, y, z)$  hacia  $f(x_0, y_0, z_0)$ . Se plantea que el cuadricóptero maestro sea el punto  $f(x_0, y_0, z_1)$  y el esclavo sea el punto  $f(x_0, y_0, z_0)$ , en la figura 3.6, se busca obtener que la nueva posición del esclavo sea  $x_d$  y este mantenga la distancia deseada con el maestro.

Como se busca que el esclavo siga al maestro, se modifica la ecuación 3.42 para que calcule  $x_d$  desde la posición del maestro y no desde el esclavo.

$$x_{ac} = x_{m_{xyz}} - \frac{d}{|x_{m_{xyz}} \ x_{s_{xyz}}|}(x_{m_{xyz}} - x_{s_{xyz}}) \quad (3.43)$$

Donde  $x_{ac} = (x_{ac_x}, x_{ac_y}, x_{ac_z})$  son los puntos deseados para el esclavo,  $d$  es la distancia deseada entre el esclavo y el maestro,  $x_{s_{xyz}}$  son las coordenadas actuales del esclavo y  $x_{m_{xyz}}$  son las coordenadas actuales del maestro.

Al implementarse la ecuación en el sistema del cuadricóptero, se observa que el esclavo mantiene la distancia deseada de manera constante y este avanza sin pararse o adelantarse al maestro.

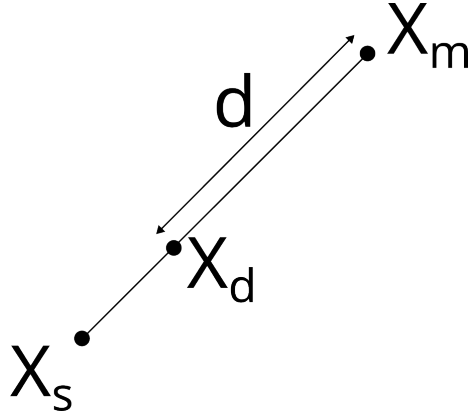


Figura 3.6: Representación de la posición deseada entre el esclavo y maestro.

El cálculo anti-colisión se aplica directamente en el sistema esclavo, donde los puntos deseados son directamente aplicados en el subsistema de acoplamiento:

$$\begin{aligned} \dot{h}_{1x} &= -\alpha h_{1x} + h_{2x} \\ \dot{h}_{2x} &= -\gamma_1 h_{1x} - \gamma_2 h_{2x} - k(x_{acx} - x_{s9}) \end{aligned} \quad (3.44)$$

$$\begin{aligned} \dot{h}_{1y} &= -\alpha h_{1y} + h_{2y} \\ \dot{h}_{2y} &= -\gamma_1 h_{1z} - \gamma_2 h_{2x} - k(x_{acy} - x_{s11}) \end{aligned} \quad (3.45)$$

$$\begin{aligned} \dot{h}_{1z} &= -\alpha h_{1z} + h_{2z} \\ \dot{h}_{2z} &= -\gamma_1 h_{1z} - \gamma_2 h_{2x} - k(x_{acz} - x_{s7}) \end{aligned} \quad (3.46)$$

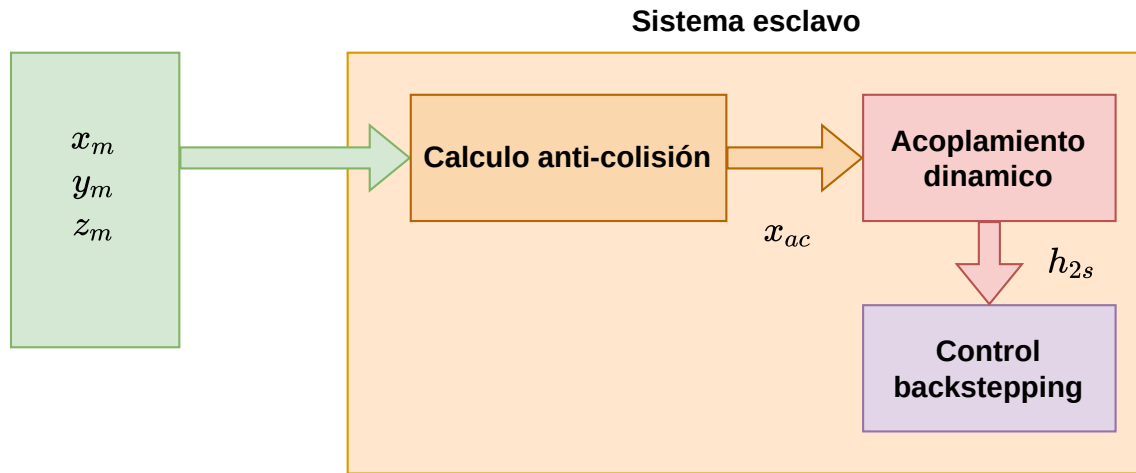


Figura 3.7: Sistema esclavo del cuadricóptero con el subsistema de anti-colisión incluido.

Igualmente, se puede visualizar el cálculo anti-colisión como un subsistema más dentro del sistema esclavo como se puede observar en la figura 3.7.

En definitiva, el cálculo anti-colisión desarrollado es una forma efectiva de mantener la distancia entre los dos cuadricópteros sin la necesidad de aplicar más sensores o hacer uso de cálculos muchos más complejos. Esto además proporciona que el cuadricóptero esclavo se mantenga a una velocidad constante sin la necesidad de realizar paradas continuas.

# Capítulo 4

## Esfera Caótica

En este capítulo se desarrolla el concepto detrás de la esfera caótica, su generación e implementación para la creación de rutas caóticas.

### 4.1. Generación de la esfera

Los sistemas caóticos convencionales tienen la peculiaridad de generar formas fractales muy características cuando son graficados con respecto a sus variables, el más conocido es el atractor de Lorenz que nos da la forma de una mariposa [14], así como su variante optimizada MACM que genera una forma similar [15], el atractor de Rössler que genera una sola espiral similar a una herradura [16], y el atractor de Aizawa que genera un toro caótico [17].

Aunque es de especial interés controlar la forma en que se genera el caos para obtener formas deseadas usando restricciones, como el atractor caótico Rössler con restricciones que genera una forma cuadrada [18]. Esto nos da la posibilidad de crear rutas caóticas que pueden ser usados en múltiples aplicaciones, como localizar objetivos, seguimiento y esquivar amenazas.

Se expone la necesidad de generar una esfera caótica en el espacio 3D como una ruta que genere internamente el sistema maestro, de esta forma se evita la necesidad de requerir de un controlador externo que envíe la señal externa, con lo que, el generador de rutas se puede incluir como un subsistema extra dentro del cuadricóptero maestro.

Para la generación de caos con características específicas se requiere la utilización de un sistema híbrido no continuo, llamados sistemas conmutados, los cuales se

pueden construir y diseñar fácilmente y generan atractores con bastante facilidad, y que pueden alimentar a ecuaciones paramétricas para generar formas deseadas.

#### 4.1.1. Atractor esférico

El atractor esférico es realizado mediante el uso de un sistema híbrido [19]:

$$\begin{aligned}\dot{r} &= -ar + af + \sin(\theta) \\ \dot{\theta} &= b \operatorname{sgn}(\cos(\theta)) + c \\ \dot{\psi} &= d \operatorname{sgn}(\sin(\theta) \sin(\psi)) + e\end{aligned}\tag{4.1}$$

Donde  $r$ ,  $\theta$  y  $\psi$  son las variables de estado,  $a, b, c, d, e$  son constantes. La función de signo es representada por  $\operatorname{sgn}$ .

El sistema caótico usa coordenadas esfericas, por lo que para convertirlo a coordenadas cartesianas se utiliza la siguiente ecuación:

$$\begin{aligned}x &= (r \sin(\psi) + r_0) \cos(\theta) \\ y &= (r \sin(\psi) + r_0) \sin(\theta) \\ z &= r \cos(\psi)\end{aligned}\tag{4.2}$$

Para obtener el atractor caótico, los parámetros  $a, b, c, d, e$  requieren satisfacer las siguientes condiciones:

$$\begin{aligned}c^2 &> b^2 \\ e^2 &> d^2 \\ b &\neq d \text{ ó } c \neq e \\ a &> 0\end{aligned}\tag{4.3}$$

Las condiciones  $c^2 > b^2$  y  $e^2 > d^2$  aseguran que las variables de estado  $\theta$  y  $\phi$  se mantengan creciendo o decreciendo conforme  $t \rightarrow \infty$  y la condición  $b \neq d$  ó  $c \neq e$  evita que el sistema genere la condición de ciclo límite.

Con lo anterior, se puede generar una esfera caótica con los siguientes valores en los parámetros:

$$a = 1000, b = 4.5, c = 7, d = -1.5, e = 3.9, f = 3.\tag{4.4}$$

Como se puede verificar en la figura 4.1 se crea una esfera con un radio de 3 unidades de distancia, donde las trayectorias resultan ser caóticas. El comportamiento del sistema resulta ser satisfactorio para que este sea usado en la generación de rutas para el cuadricóptero.

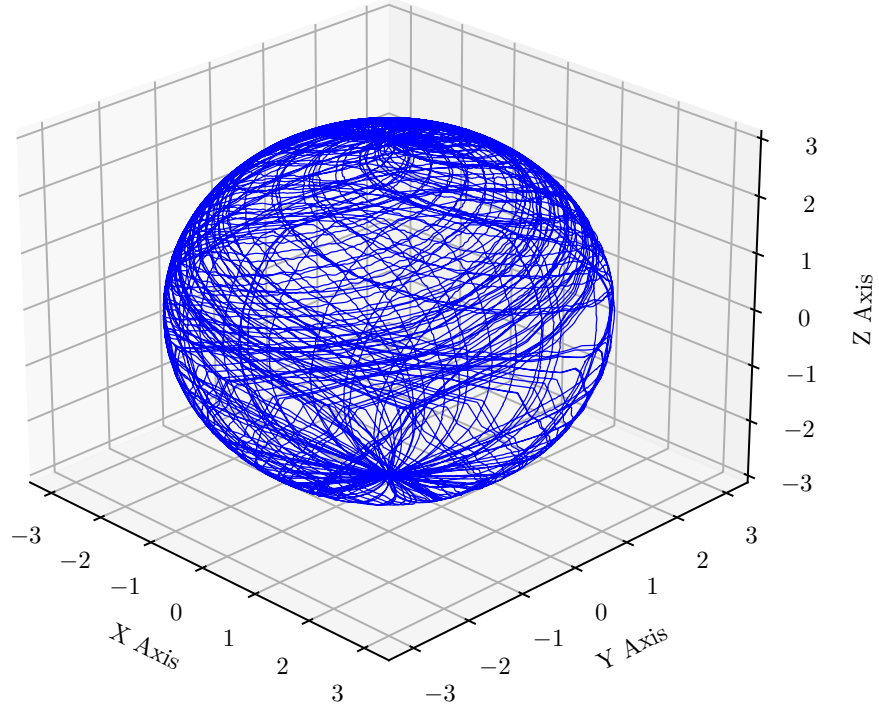


Figura 4.1: Esfera caótica generada utilizando las constantes 4.4.

## 4.2. Implementación

Para la generación de rutas utilizando el sistema caótico esférico, este se reduce a solo dos variables de estados los cuales son  $\theta$  y  $\psi$ , la variable de estado  $r$  pasa a ser una constante. De esta forma se tiene el sistema dinámico original 4.1 reducido:

$$\begin{aligned}\dot{\theta} &= b \operatorname{sgn}(\cos(\psi)) + c \\ \dot{\psi} &= d \operatorname{sgn}(\sin(\theta) \sin(\psi)) + e\end{aligned}\tag{4.5}$$

Donde los valores de  $\phi$  y  $\theta$  son usados para generar la esfera usando ecuaciones paramétricas:

$$\begin{aligned}x_d &= x_0 + r \sin(\psi) \cos(\theta) \\ y_d &= y_0 + r \sin(\psi) \sin(\theta) \\ z_d &= z_0 + r \cos(\psi)\end{aligned}\tag{4.6}$$

Para la generación del sistema 4.5, se utiliza el método de Euler [20] para resolver el sistema. Esto nos da como beneficio de que el cuadricóptero no invierta mucho tiempo calculando el siguiente punto a seguir.

De esta forma se tiene el siguiente pseudocódigo:

Algoritmo 4.1: Pseudocódigo sobre la implementación para resolver el sistema caótico

---

```

variable h, xyz_tolerancia, compensacion_z, xyz_actuales[3],
        xyz_deseados[3], valores_caoticos_anteriores[2],
        valores_caoticos_actuales[2], paso[2]
h = 0.01

xyz_actuales[3] = esfera(valores_caoticos_anteriores, compensacion_z)

Si abs(xyz_actuales - posicion_actual_del_cuadricoptero) < xyz_tolerancia
    paso[2] = sistema_caotico(valores_caoticos_anteriores)
    valores_caoticos_actuales[2] = valores_caoticos_anteriores + h * paso
    xyz_actuales[3] = esfera(valores_caoticos_actuales, compensacion_z)
Fin Si

valores_caoticos_anteriores[2] = valores_caoticos_actuales
xyz_deseados[3] = xyz_actuales[3]

```

---

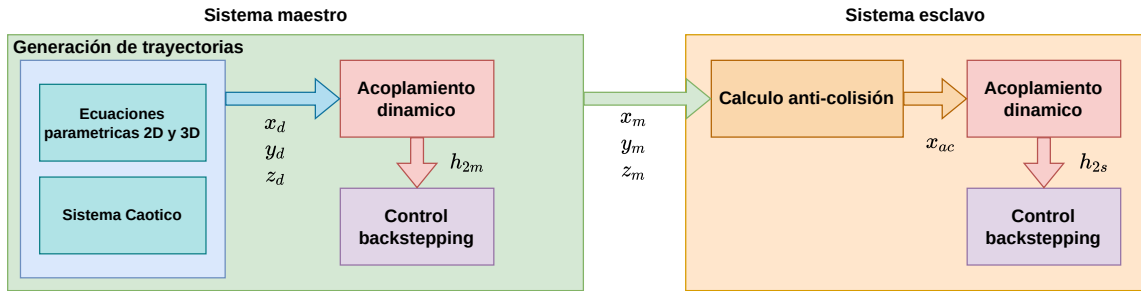


Figura 4.2: Organización de los sistemas y sus interacciones.

Como se puede ver en la figura 4.2, el cuadricóptero maestro formara las rutas, mientras que el esclavo solamente seguirá la ruta del maestro usando el acoplamiento dinámico descrito anteriormente, haciendo que los cuadricópteros no dependan de un control externo para generar la ruta que se le indique.

Se concluye que el sistema caótico de la esfera cumple con los requisitos que se buscan para generar trayectorias caóticas con el cuadricóptero, además de que se pueden crear otras figuras de tipo esféricas usando el mismo sistema caótico solamente cambiando las ecuaciones paramétricas. De modo similar, el atractor Aizawa [17] que genera una “esfera”, usando un conjunto de ecuaciones que empieza a oscilar y tener flujo turbulento debido a las interacciones entre la bifurcación de Hopf y de histéresis. Con los correctos valores en los parámetros, empieza a tener un comportamiento caótico similar a la esfera investigada.

A pesar de la similitud en estos sistemas dinámicos, son totalmente diferentes, puesto que el sistema caótico de la esfera investigada nos proporciona más control sobre la esfera generada debido usa coordenadas esféricas y permite más flexibilidad a la hora de convertirlo a coordenadas cartesianas; de tal forma, se puede generar toros, flores, etc. Mientras que el atractor Aizawa trabaja directamente en el plano cartesiano, por otro lado las ecuaciones no están completamente estudiadas y su comportamiento global es aún desconocido.

# Capítulo 5

## Resultados

En este capítulo se presentan los resultados computacionales que se obtuvieron, mostrando las diferentes trayectorias y los comportamientos de los cuadricópteros acoplados en la configuración maestro-esclavo.

### 5.1. Detalles computacionales

Para la solución del sistema dinámico se utilizó el software de MATLAB 2021a y la función ODE45 para resolver las ecuaciones de estado del sistema dinámico, se utilizó Python y la librería matplotlib para la creación de las gráficas con base en los resultados de MATLAB.

Para las simulaciones se utilizaron los siguientes valores iniciales para el maestro:  $x_{m_5} = \psi_0 = 0$ ,  $x_{m_9} = x_0 = 0$ ,  $x_{m_{11}} = y_0 = 0$  y  $x_{m_7} = z_0 = 0$ . Y para el sistema esclavo los siguientes valores:  $x_{s_5} = \psi_0 = 0$ ,  $x_{s_9} = x_0 = 2$ ,  $x_{s_{11}} = y_0 = 2$  y  $x_{s_7} = z_0 = 0$ . De igual modo se utiliza la ley de control diseñada previamente en el capítulo anterior, donde el código desarrollado para realizar las simulaciones se puede encontrar en el apéndice A.

## 5.2. Trayectorias en dos dimensiones

Para las trayectorias en dos dimensiones se utilizaron diferentes figuras definidas por funciones matemáticas, mientras que para el valor de Z se utilizó un número fijo.

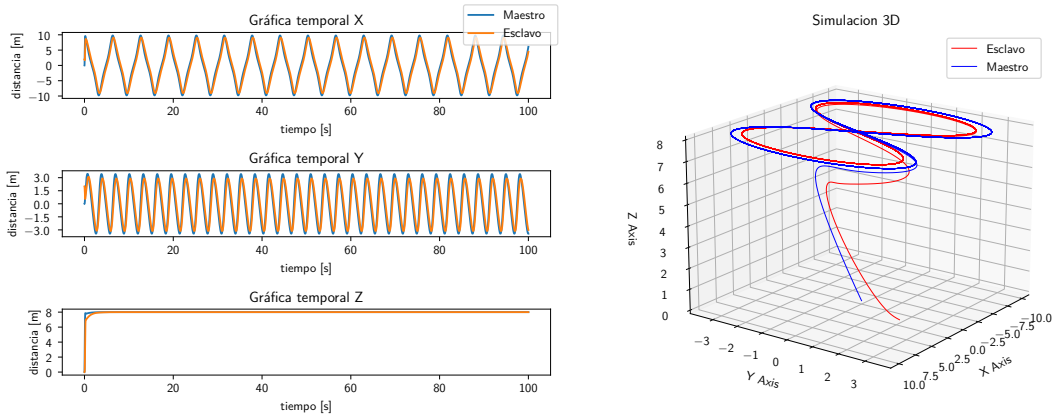
### 5.2.1. Trayectoria Lemniscata de Bernoulli

La trayectoria de la lemniscata de Bernoulli está representada por las siguientes ecuaciones paramétricas.[\[21\]](#)

$$\begin{aligned}x_d &= \frac{a \cos(t)}{1 + \sin^2(t)} \\y_d &= \frac{a \sin(t) \cos(t)}{1 + \sin^2(t)} \\z_d &> 0\end{aligned}\tag{5.1}$$

Donde la  $x_d$ ,  $y_d$  y  $z_d$  son las coordenadas cartesianas que tomara el cuadricóptero,  $a$  es una constante positiva que define la distancia desde el centro a uno de los extremos.

Los ángulos de balanceo, guiñada y cabeceo deseados son  $\phi = 0$ ,  $\psi = 0$  y  $\theta = 0$ . En la figura 5.1b se muestra la simulación de los dos cuadricópteros y en la figura 5.1a se muestran las gráficas temporales para el sistema maestro y esclavo. Como se puede observar el cuadricóptero maestro sigue la ruta deseada y el esclavo sigue al maestro manteniendo una distancia específica.



(a) Respuesta temporal.

(b) Resultados de la simulación.

Figura 5.1: Trayectoria de los cuadricópteros: Lemniscata.

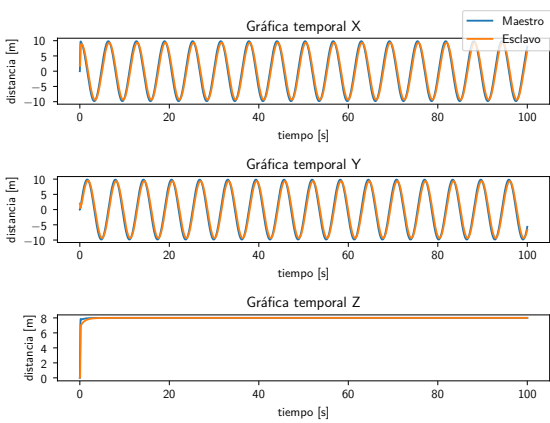
### 5.2.2. Trayectoria circular

La trayectoria circular está representada por las siguientes ecuaciones paramétricas.[21]

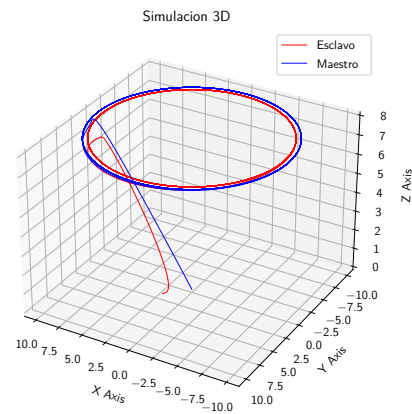
$$\begin{aligned}x_d &= a \cos(t) \\y_d &= a \sin(t) \\z_d &> 0\end{aligned}\tag{5.2}$$

Donde la  $x_d$ ,  $y_d$  y  $z_d$  son las coordenadas cartesianas que tomara el cuadricóptero,  $a$  es el radio del círculo.

En la figura 5.2a y 5.2b se muestran los resultados obtenidos en la simulación para el maestro y el esclavo.



(a) Respuesta temporal.



(b) Resultados de la simulación.

Figura 5.2: Trayectoria de los cuadricópteros: Circular.

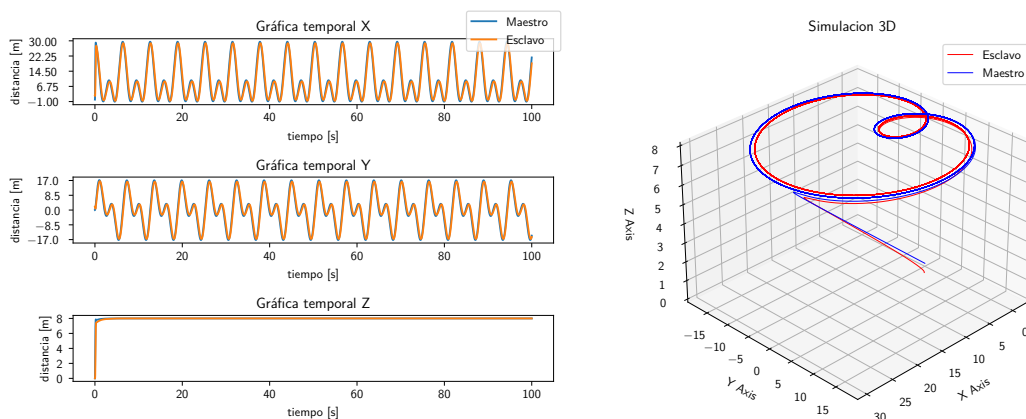
### 5.2.3. Trayectoria trisectriz caracol

La trayectoria trisectriz está representada por las siguientes ecuaciones paramétricas.[21]

$$\begin{aligned}x_d &= \cos(t) (2 a \cos(t) + b) \\y_d &= \sin(t) (2 a \cos(t) + b) \\z_d &> 0\end{aligned}\tag{5.3}$$

Donde la  $x_d$ ,  $y_d$  y  $z_d$  son las coordenadas cartesianas que tomara el cuadricóptero,  $a$  es el radio de la trisectriz,  $b$  es el parámetro de la conoide. El valor de  $a$  y  $b$  deben ser iguales para obtener la trisectriz caracol.

En la figura 5.2a y 5.2b se muestran los resultados obtenidos en la simulación para el maestro y el esclavo.



(a) Respuesta temporal.

(b) Resultados de la simulación.

Figura 5.3: Trayectoria de los cuadricópteros: Trisectriz.

## 5.3. Trayectorias en tres dimensiones

### 5.3.1. Trayectoria Esférica

La trayectoria esférica está realizada de una forma similar a la que se generan los vértices de un objeto esférico en computadora.[22] Sin embargo, se modificó el algoritmo para que se creen los puntos de forma continua y sin necesidad de almacenar todos los puntos de la esfera en la memoria.

Para la generación de las coordenadas deseadas, se utiliza la ecuación paramétrica de la esfera para obtener los puntos requeridos.

$$\begin{aligned}x_d &= x_0 + r \sin(\phi) \cos(\theta) \\y_d &= y_0 + r \sin(\phi) \sin(\theta) \\z_d &= z_0 + r \cos(\phi)\end{aligned}\tag{5.4}$$

Donde  $x_0$ ,  $y_0$  y  $z_0$  es la posición del centro de la esfera,  $r$  es el radio,  $0 > \phi > \pi$  y  $0 > \theta > 2\pi$ .

Como se puede observar en la ecuación 5.4, se requiere de  $\phi$  y  $\theta$  para generar los puntos en la esfera que el cuadricóptero requiere seguir, primeramente se observa que  $\phi$  modifica la posición vertical y  $\theta$  la posición horizontal. De esta manera para hacer más sencillo el cálculo de los vértices para general la esfera se divide en “stacks” y “slices”, los cuales el “stack” son las divisiones verticales y los “slices” las divisiones en horizontal como se puede ver en la figura 5.4. Cada “stack” y “slice” sería un punto en la superficie de la esfera. Por lo que definimos la cantidad máxima de “stacks” y “slices” para calcular los ángulos e introducirlos en las ecuaciones paramétricas y obtener los puntos deseados para trazar una ruta. Las ecuaciones para generar  $\phi$  y  $\theta$  serían las siguientes:

$$\theta = 2\pi \cdot \frac{\text{sliceStep}}{\text{sliceCount}}\tag{5.5}$$

Donde  $2\pi$  es el valor máximo de  $\theta$ ,  $\text{sliceStep}$  es una variable que aumenta por uno a medida que se calcula cada punto, y  $\text{sliceCount}$  es la cantidad máxima definida de “slices”.

$$\phi = \pi \cdot \frac{\text{stackStep}}{\text{stackCount}}\tag{5.6}$$

Donde  $\pi$  es el valor máximo de  $\phi$ ,  $\text{stackStep}$  es una variable que aumenta por uno a medida que se calcula cada punto, y  $\text{stackCount}$  es la cantidad máxima definida de “stacks”.

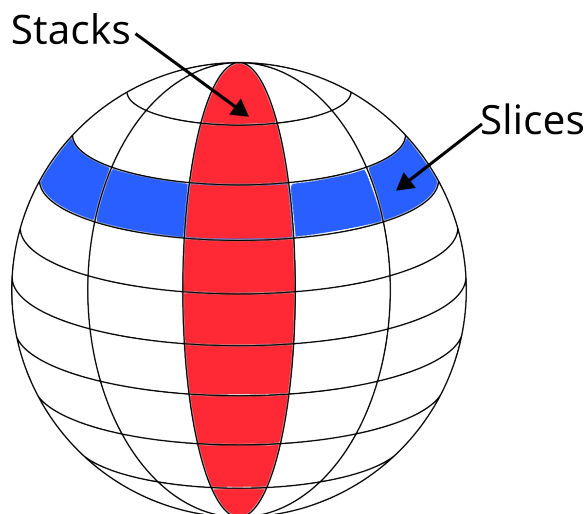
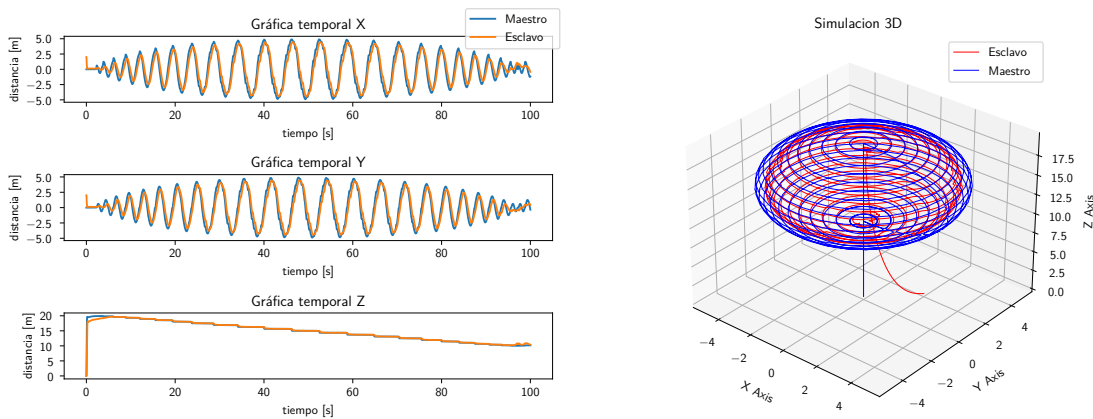


Figura 5.4: Divisiones de la esfera.

Dado a lo anterior se puede generar una esfera de manera continua, para crear la esfera los valores utilizados son  $stackCount = 24$  y  $sliceCount = 30$ . En las figuras 5.5a y 5.5b se muestran los resultados de la simulación de los cuadricópteros maestro-esclavo.



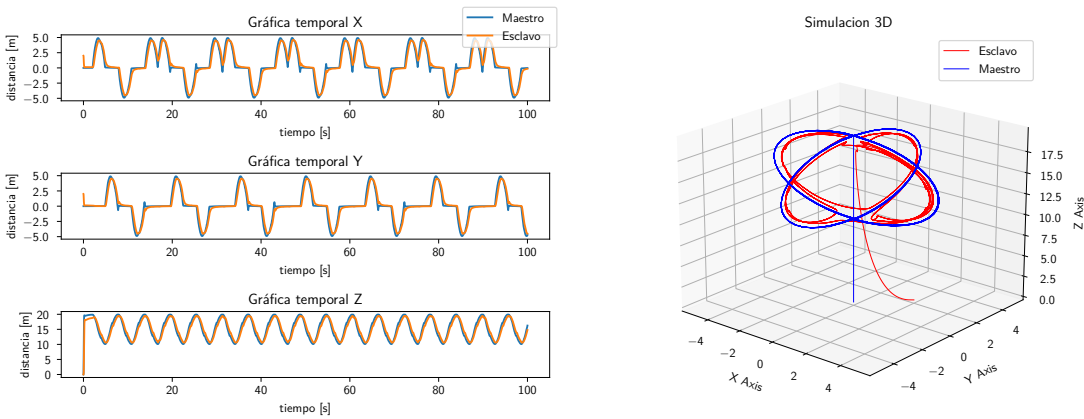
(a) Respuesta temporal.

(b) Resultados de la simulación.

Figura 5.5: Trayectoria de los cuadricópteros: Esférica.

### 5.3.2. Trayectoria de media luna

La trayectoria de media luna está basada en la trayectoria esférica que se vio anteriormente, con modificaciones en el algoritmo de generación de “stacks” y “slices” para que estos se generen de manera vertical en vez de manera horizontal. Para la generación de las medias lunas los valores utilizados son  $stackCount = 20$  y  $sliceCount = 4$  obteniendo así los resultados que se pueden observar en las figuras 5.6a y 5.6b.



(a) Respuesta temporal.

(b) Resultados de la simulación.

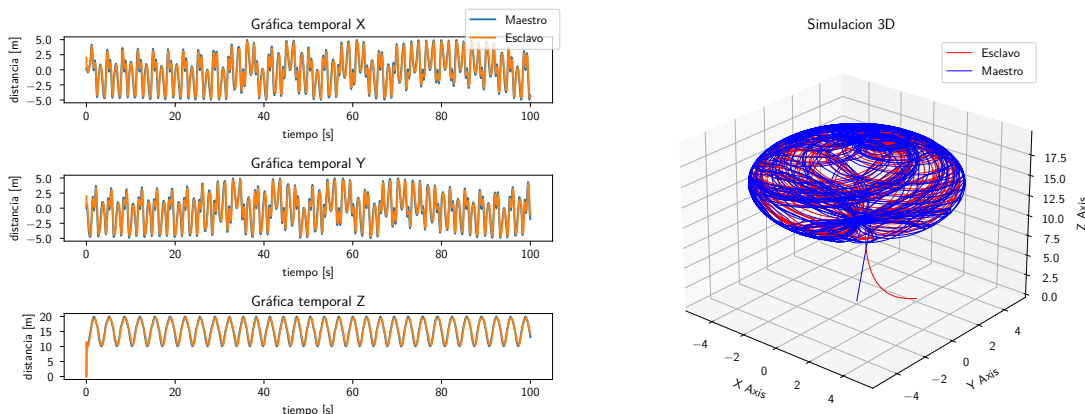
Figura 5.6: Trayectoria de los cuadricópteros: Media Luna.

## 5.4. Trayectorias caóticas

### 5.4.1. Esfera caótica

Para la trayectoria de la esfera caótica, se utiliza las ecuaciones del sistema dinámico 4.5. El cuadricóptero utiliza el método de Euler para generar el sistema caótico y se utiliza la ecuación paramétrica de la esfera para generar los puntos  $X$ ,  $Y$  y  $Z$ . 4.6

Los valores iniciales que se tomaron para el sistema caótico fueron  $\phi_0 = 1$ ,  $\theta_0 = 3$ . En las figuras 5.7a y 5.7b se pueden observar los resultados de la simulación.



(a) Respuesta temporal. (b) Resultados de la simulación.

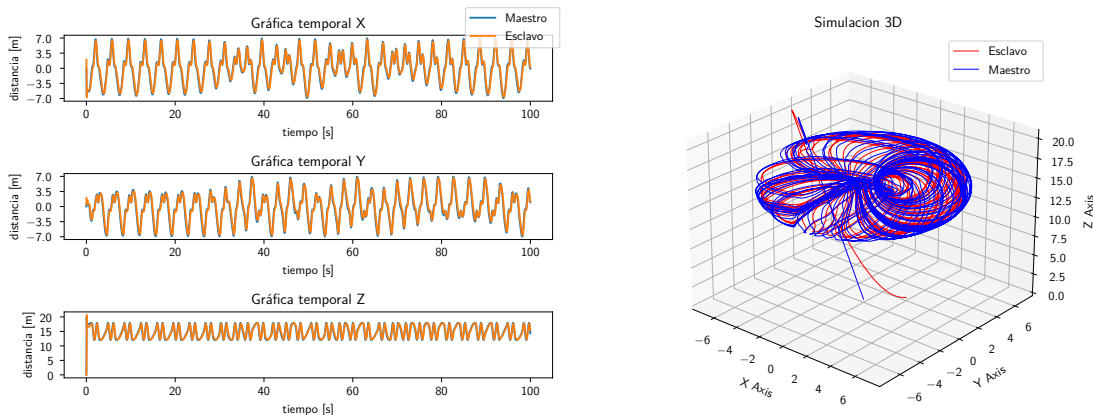
Figura 5.7: Trayectoria de los cuadricópteros: Esfera caótica.

### 5.4.2. Toro caótico

Para la trayectoria del toro caótico, se reutilizo el sistema caótico de la esfera caótica. Esta vez usando la ecuación paramétrica del toro.

$$\begin{aligned}
 x_d &= (R + r \cos(\theta)) \cos(\psi) \\
 y_d &= (R + r \cos(\theta)) \sin(\psi) \\
 z_d &= z_0 + r \sin(\theta)
 \end{aligned}
 \tag{5.7}$$

Donde  $R$  es la distancia del centro del conducto hacia el centro del toro,  $r$  es el radio del conducto. El valor de  $R$  y  $r$  son constantes con  $r < R$ . Para la simulación se usaron los siguientes valores  $r = 3$  y  $R = 4$ . En las figuras 5.8a y 5.8b se pueden observar los resultados de la simulación.



(a) Respuesta temporal. (b) Resultados de la simulación.

Figura 5.8: Trayectoria de los cuadricópteros: Toro caótico.

# Capítulo 6

## Conclusiones

En esta tesis de licenciatura se realizó el control backstepping en una red de dos cuadricópteros en configuración maestro-esclavo con el fin de realizar el seguimiento de trayectorias, en particular el seguimiento de trayectorias caóticas esféricas. Como se pudo observar en la sección 3, con el control backstepping aplicado al cuadricóptero, se logró mantener estabilidad y alcanzar el seguimiento de la trayectoria deseada de acuerdo a la serie de puntos que fueron insertados al control.

Los resultados en cuanto a la sincronización y formación de los cuadricópteros fueron los esperados, donde mediante un acoplamiento dinámico unidireccional y con una fuerza de acoplamiento seleccionada adecuadamente, se pudo observar que en todos los casos el esclavo lograba la formación con el maestro. Aunque cabe destacar que en las etapas tempranas de pruebas el esclavo se adelantaba o colisionaba ocasionalmente con el maestro, por lo que para solucionar el problema se agregó el cálculo anti-colisión para evitar de manera activa la colisión entre los dos cuadricópteros.

Las trayectorias que realizaron los cuadricópteros fueron variadas, las cuales se pueden clasificar en 3 partes: Trayectorias en el espacio 2D, Trayectorias en el espacio 3D y la trayectoria caótica. La trayectoria en el espacio 2D fue realizada con el fin de hacer pruebas sobre el modelo de control, de lo cual obtuvimos algunos resultados preliminares esperados.

En el caso seguimiento de las trayectorias en el espacio 3D los resultados fueron satisfactorios donde se adaptaron los algoritmos de generación de gráficos por computadora a una forma continua.

Finalmente, la trayectoria caótica esférica se logró de forma exitosa, donde usando la integración de Euler dentro del maestro fue posible generar los puntos para

seguir la ruta caótica. Un caso a destacar fue la generación del toro caótico, donde solamente fue requerido intercambiar las ecuaciones paramétricas de la esfera por las del toro, lo que resalta la facilidad de adaptabilidad de la metodología presentada.

Por lo tanto, se logró alcanzar la sincronización y formación entre los dos cuadricópteros en configuración maestro-esclavo en una trayectoria caótica esférica.

## 6.1. Trabajo Futuro

A futuro se plantea realizar experimentos en el laboratorio con el fin de comprobar la implementación del sistema, así como verificar exitosamente el funcionamiento del modelo de control propuesto y poder comprobarlo con los resultados obtenidos en la simulación.

Así mismo se busca implementar el uso de la esfera caótica en una tarea en específico, esto puede ser de búsqueda, transporte, seguimiento, patrullaje, entre otros.

Agregar más cuadricópteros para al modelo de control general de un solo maestro y múltiples esclavos.

Hacer que los cuadricópteros sean conscientes de su entorno para evitar colisionar con elementos externos.

# Bibliografía

- [1] V. Sundarapandian y Ahmad Taher Azar, eds. *Backstepping Control of Nonlinear Dynamical Systems*. Advances in Nonlinear Dynamics and Chaos Series. London, United Kingdom : San Diego, CA, United States: Academic Press, an imprint of Elsevier, 2021. 517 págs. ISBN: 978-0-12-817582-8.
- [2] J. Pena Ramirez, A. Arellano-Delgado y H. Nijmeijer. «Enhancing Master-Slave Synchronization: The Effect of Using a Dynamic Coupling». En: *Physical Review E* 98.1 (13 de jul. de 2018), pág. 012208. DOI: [10.1103/PhysRevE.98.012208](https://doi.org/10.1103/PhysRevE.98.012208). URL: <https://link.aps.org/doi/10.1103/PhysRevE.98.012208>.
- [3] B. P. Lathi y R. A. Green. *Linear Systems and Signals*. Third Edition. The Oxford Series in Electrical and Computer Engineering. New York: Oxford University Press, 2018. 988 págs. ISBN: 978-0-19-020017-6.
- [4] Hassan K. Khalil. *Nonlinear Systems*. 3rd ed. Upper Saddle River, N.J: Prentice Hall, 2002. 750 págs. ISBN: 978-0-13-067389-3.
- [5] Yaneer Bar-Yam. *Dynamics of Complex Systems*. Studies in Nonlinearity. Reading, Mass: Addison-Wesley, 1997. 848 págs. ISBN: 978-0-201-55748-0.
- [6] Y. Tang, A. Mees y L. Chua. «Synchronization and Chaos». En: *IEEE Transactions on Circuits and Systems* 30.9 (sep. de 1983), págs. 620-626. ISSN: 1558-1276. DOI: [10.1109/TCS.1983.1085409](https://doi.org/10.1109/TCS.1983.1085409). URL: <https://ieeexplore.ieee.org/document/1085409>.
- [7] Reggie Brown, Nikolai F. Rulkov y Eugene R. Tracy. «Modeling and Synchronizing Chaotic Systems from Time-Series Data». En: *Physical Review E* 49.5 (1 de mayo de 1994), págs. 3784-3800. DOI: [10.1103/PhysRevE.49.3784](https://doi.org/10.1103/PhysRevE.49.3784). URL: <https://link.aps.org/doi/10.1103/PhysRevE.49.3784>.
- [8] M. Sami Fadali y Antonio Visioli. *Digital Control Engineering: Analysis and Design*. Third Edition. United Kingdom: Elsevier/Academic press is an imprint of Elsevier, 2020. 678 págs. ISBN: 978-0-12-814433-6.

- [9] J. a. K. Suykens, P. F. Curran y L. O. Chua. «Master-Slave Synchronization Using Dynamic Output Feedback». En: *International Journal of Bifurcation and Chaos* 07.03 (mar. de 1997), págs. 671-679. ISSN: 0218-1274. DOI: [10.1142/S0218127497000467](https://doi.org/10.1142/S0218127497000467). URL: <https://www.worldscientific.com/doi/abs/10.1142/S0218127497000467>.
- [10] Hai Lin y Panos J. Antsaklis. *Hybrid Dynamical Systems: Fundamentals and Methods*. Advanced Textbooks in Control and Signal Processing. Cham: Springer International Publishing, 2022. ISBN: 978-3-030-78729-5 978-3-030-78731-8. DOI: [10.1007/978-3-030-78731-8](https://doi.org/10.1007/978-3-030-78731-8). URL: <https://link.springer.com/10.1007/978-3-030-78731-8>.
- [11] S. Bouabdallah y R. Siegwart. «Backstepping and Sliding-mode Techniques Applied to an Indoor Micro Quadrotor». En: *Proceedings of the 2005 IEEE International Conference on Robotics and Automation*. Proceedings of the 2005 IEEE International Conference on Robotics and Automation. Abr. de 2005, págs. 2247-2252. DOI: [10.1109/ROBOT.2005.1570447](https://doi.org/10.1109/ROBOT.2005.1570447).
- [12] Enrica Soria, Fabrizio Schiano y Dario Floreano. «Predictive Control of Aerial Swarms in Cluttered Environments». En: *Nature Machine Intelligence* 3.6 (6 jun. de 2021), págs. 545-554. ISSN: 2522-5839. DOI: [10.1038/s42256-021-00341-y](https://doi.org/10.1038/s42256-021-00341-y). URL: <https://www.nature.com/articles/s42256-021-00341-y>.
- [13] Feifei Zhao et al. «Nature-Inspired Self-Organizing Collision Avoidance for Drone Swarm Based on Reward-Modulated Spiking Neural Network». En: *Patterns* 3.11 (11 de nov. de 2022), pág. 100611. ISSN: 2666-3899. DOI: [10.1016/j.patter.2022.100611](https://doi.org/10.1016/j.patter.2022.100611). URL: <https://www.sciencedirect.com/science/article/pii/S2666389922002367>.
- [14] Pierre Bergé, Yves Pomeau y C. Vidal. *Order within chaos: towards a deterministic approach to turbulence*. New York : Paris: Wiley ; Hermann, 1986. 329 págs. ISBN: 978-0-471-84967-4.
- [15] Rodrigo Méndez-Ramírez et al. «A New Simple Chaotic Lorenz-Type System and Its Digital Realization Using a TFT Touch-Screen Display Embedded System». En: *Complexity* 2017 (26 de jul. de 2017), e6820492. ISSN: 1076-2787. DOI: [10.1155/2017/6820492](https://doi.org/10.1155/2017/6820492). URL: <https://www.hindawi.com/journals/complexity/2017/6820492/>.
- [16] O. E. RöSSLer. «An Equation for Continuous Chaos». En: *Physics Letters A* 57.5 (12 de jul. de 1976), págs. 397-398. ISSN: 0375-9601. DOI: [10.1016/0375-9601\(76\)90101-8](https://doi.org/10.1016/0375-9601(76)90101-8). URL: <https://www.sciencedirect.com/science/article/pii/0375960176901018>.

- [17] W. F. Langford. «Numerical Studies of Torus Bifurcations». En: *Numerical Methods for Bifurcation Problems: Proceedings of the Conference at the University of Dortmund, August 2226, 1983*. Ed. por T. Küpper, H. D. Mittelmann y H. Weber. International Series of Numerical Mathematics / Internationale Schriftenreihe Zur Numerischen Mathematik / Série Internationale d'Analyse Numérique. Basel: Birkhäuser, 1984, págs. 285-295. ISBN: 978-3-0348-6256-1. DOI: [10.1007/978-3-0348-6256-1\\_19](https://doi.org/10.1007/978-3-0348-6256-1_19). URL: [https://doi.org/10.1007/978-3-0348-6256-1\\_19](https://doi.org/10.1007/978-3-0348-6256-1_19).
- [18] Qingfei Chen et al. «Chaotic Attractors in Striped Rectangular Shapes Generated by a Rössler-like System». En: *Physics Letters A* 348.3 (2 de ene. de 2006), págs. 195-200. ISSN: 0375-9601. DOI: [10.1016/j.physleta.2005.08.085](https://doi.org/10.1016/j.physleta.2005.08.085). URL: <https://www.sciencedirect.com/science/article/pii/S037596010501354X>.
- [19] Qingfei Chen, Yiguang Hong y Guanrong Chen. «Chaotic Behaviors and Toroidal/Spherical Attractors Generated by Discontinuous Dynamics». En: *Physica A: Statistical Mechanics and its Applications* 371.2 (15 de nov. de 2006), págs. 293-302. ISSN: 0378-4371. DOI: [10.1016/j.physa.2006.03.047](https://doi.org/10.1016/j.physa.2006.03.047). URL: <https://www.sciencedirect.com/science/article/pii/S037843710600389X>.
- [20] James F. Epperson. *An Introduction to Numerical Methods and Analysis*. Second edition. Hoboken, New Jersey: John Wiley & Sons, Inc, 2013. 591 págs. ISBN: 978-1-118-36759-9.
- [21] J. Dennis Lawrence. *A Catalog of Special Plane Curves*. New York: Dover Publications, 1972. 218 págs. ISBN: 978-0-486-60288-2.
- [22] FreeDesktop Organization. *Mesa / Glu*. FreeDesktop Organization, 13 de ene. de 2023. URL: <https://gitlab.freedesktop.org/mesa/glu>.
- [23] Rolando Diaz Castillo. «Vuelo en formacion de vehículos aéreos no tripulados». Ensenada, Baja California: CICESE, 2018. 100 págs.
- [24] Johnny Rubio Pecasso. «Formación de Cuadricópteros Utilizando Un Control Por Backstepping y Un Acoplamiento Dinámico». Ensenada, Baja California: Universidad Autónoma de Baja California, 2022. 153 págs.
- [25] Hicham Khebbache y M. Tadjine. «Robust Fuzzy Backstepping Sliding Mode Controller For a Quadrotor Unmanned Aerial Vehicle». En: *Control Engineering and Applied Informatics* 15 (1 de jul. de 2013), págs. 3-11.

- [26] Abdul-Wahid A. Saif et al. «Modified Backstepping Control of Quadroter». En: *International Multi-Conference on Systems, Signals & Devices*. International Multi-Conference on Systems, Signals & Devices. Mar. de 2012, págs. 1-6. DOI: [10.1109/SSD.2012.6197975](https://doi.org/10.1109/SSD.2012.6197975).

# Apéndice A

## Código para la simulación del control de los cuadricópteros y sus configuraciones

---

```
%  
% Control de Drone, maestro-esclavo con con diferentes rutas.  
%  
% Realizado por Alejandro Duran con ayuda de codigos  
% proporcionados por el asesor.  
%  
  
clear all;  
close all;  
clc;  
  
% Variable globales  
  
global g_xyz;  
global g_seleccion_ruta;  
global g_dist_drone;  
g_xyz = 0;  
  
g_dist_drone = 0.5; % Distancia deseada en metros entre los cuadricopteros  
  
% Inicio del programa.
```



```

0; 0; 0; 0; 0; 0; 0; 0; 0; 2; 0; 2; 0; % Sistema esclavo.
0; 0; 0; 0; 0; 0; % Acoplamiento dinamico maestro.
0; 0; 0; 0; 0; 0]; % Acoplamiento dinamico esclavo.

% Tiempo de simulacion
t0 = [0 100];

tic
[t, x] = ode45(@sysdrone, t0, x0);
toc

% Graficas de tiempo.
fig1 = figure(1);
fig1.WindowState = 'maximized';

% Roll o Balanceo

subplot(3,2,1);
hold on;
plot(t, x(:,1));
plot(t, x(:,13));
grid on;
title('Grafica temporal [\phi] Roll')
xlabel('tiempo')
ylabel('Radianes')
legend('Maestro', 'Esclavo')

subplot(3,2,3);
hold on;
plot(t, x(:,3));
plot(t, x(:,15));
grid on;
title('Grafica temporal [\theta] Pitch')
xlabel('tiempo')
ylabel('Radianes')
legend('Maestro', 'Esclavo')

subplot(3,2,5);
hold on;
plot(t, x(:,5));
plot(t, x(:,17));

```

```
grid on;
title('Grafica temporal [\psi] Yaw')
xlabel('tiempo')
ylabel('Radianes')
legend('Maestro', 'Esclavo')

subplot(3,2,2);
hold on;
plot(t, x(:,7));
plot(t, x(:,19));
grid on;
title('Grafica temporal Z')
xlabel('tiempo')
ylabel('metros')
legend('Maestro', 'Esclavo')

subplot(3,2,4);
hold on;
plot(t, x(:,9));
plot(t, x(:,21));
grid on;
title('Grafica temporal X')
xlabel('Tiempo')
ylabel('metros')
legend('Maestro', 'Esclavo')

subplot(3,2,6);
hold on;
plot(t, x(:,11));
plot(t, x(:,23));
grid on;
title('Grafica temporal Y')
xlabel('Tiempo')
ylabel('metros')
legend('Maestro', 'Esclavo')

% Simulacion
fig2 = figure(2);
fig2.WindowState = 'maximized';

hold on;
```

```

%plot3(g_xyz(1,:), g_xyz(2,:), g_xyz(3,:), '-');
plot3(x(:,9), x(:,11), x(:,7), '-');
plot3(x(:,21), x(:,23), x(:,19), '--');
grid on;
title('Simulacion')
xlabel('Eje X'); ylabel('Eje Y'); zlabel('Eje Z', 'rot',-1);
%legend('Trayectoria', 'Maestro', 'Esclavo')
legend('Maestro', 'Esclavo')

% Almacenar los resultados de simulacion para su visualizacion en python
switch g_seleccion_ruta
case 1
save('lemniscata_variables.mat', 't0', 'x0', 't','x','g_xyz', '-v7');
case 2
save('circulo_variables.mat', 't0', 'x0', 't','x','g_xyz', '-v7');
case 3
save('trisectriz_variables.mat', 't0', 'x0', 't','x','g_xyz', '-v7');
case 4
save('esfera_variables.mat', 't0', 'x0', 't','x','g_xyz', '-v7');
case 5
save('media_luna_variables.mat', 't0', 'x0', 't','x','g_xyz', '-v7');
case 6
save('esfera_caos_variables.mat', 't0', 'x0', 't','x','g_xyz', '-v7');
case 7
save('toroide_caos_variables.mat', 't0', 'x0', 't','x','g_xyz', '-v7');
end

function dx=sysdrone(t, x)
% Variables globales.
global g_xyz;
global g_seleccion_ruta;
global g_r;
global g_zoff;
global g_caos_x;
global g_dist_tol;
global g_sector;
global g_stack;
global g_flag;
global g_R;
global g_r_m;

```

```

% Cuadricoptero maestro.
xm = x(1:12); % Variables de estado del cuadricoptero.
adm = x(25:30); % Variables del acoplamiento dinamico.

% Velocidad del rotor.
wm = ones(1, 4) * 10;

% Se calcula el siguiente punto a seguir.
switch g_seleccion_ruta
case 1
xyzd = lemniscata(t);

case 2
xyzd = circulo(t);

case 3
xyzd = trisectriz(t);

case 4
[stack, sector, psi, theta, flag] = trayectoria_esfera(g_sector, g_stack,
    g_flag);
xyzc = esfera(g_r, theta, psi, g_zoff);

if abs(xyzc(3) - xm(7)) < g_dist_tol && abs(xyzc(1) - xm(9)) < g_dist_tol
    && abs(xyzc(2) - xm(11)) < g_dist_tol
g_sector = sector;
g_stack = stack;
g_flag = flag;
end

xyzd = xyzc;
case 5
[stack, sector, psi, theta, flag] = trayectoria_media_luna(g_sector,
    g_stack, g_flag);
xyzc = esfera(g_r, theta, psi, g_zoff);

if abs(xyzc(3) - xm(7)) < g_dist_tol && abs(xyzc(1) - xm(9)) < g_dist_tol
    && abs(xyzc(2) - xm(11)) < g_dist_tol
g_sector = sector;
g_stack = stack;

```

49

```
g_flag = flag;
end

xyzd = xyzc;
case 6
h = 0.01;
xyzc = esfera(g_r, g_caos_x(1), g_caos_x(2), g_zoff);

if abs(xyzc(3) - xm(7)) < g_dist_tol && abs(xyzc(1) - xm(9)) < g_dist_tol
    && abs(xyzc(2) - xm(11)) < g_dist_tol
f = gen_esfera_caotica(g_caos_x);
g_caos_x = g_caos_x + h.*f;
xyzc = esfera(g_r, g_caos_x(1), g_caos_x(2), g_zoff);
end

xyzd = xyzc;
case 7
h = 0.01;
xyzc = toroide(g_r_m, g_caos_x(1), g_caos_x(2), g_zoff, g_R);

if abs(xyzc(3) - xm(7)) < g_dist_tol && abs(xyzc(1) - xm(9)) < g_dist_tol
    && abs(xyzc(2) - xm(11)) < g_dist_tol
f = gen_esfera_caotica(g_caos_x);
g_caos_x = g_caos_x + h.*f;
xyzc = toroide(g_r_m, g_caos_x(1), g_caos_x(2), g_zoff, g_R);
end

xyzd = xyzc;
end

% Guardamos la ruta calculada.
if g_xyz == 0
g_xyz = [xyzd(1); xyzd(2); xyzd(3)];
else
g_xyz = [g_xyz(1,:), xyzd(1);
g_xyz(2,:), xyzd(2);
g_xyz(3,:), xyzd(3)];
end

% Insertamos
```

```

xdm = zeros(1, 12);
xdm(7) = xyzd(3); % Z
xdm(9) = xyzd(1); % X
xdm(11) = xyzd(2); % Y

% Ganancias
%ganancia=30;
ganancia=100; % caos
alfa = ones(1, 12) * ganancia;

% Error
z = calcular_error_z_ad(xdm, xm, alfa, adm);

% Control
dxm = Drone(xm, wm, alfa, z);

% Se envian las variables de sincronizacion.
msg_xyzm = [xm(9), xm(11), xm(7)];

% Cuadricoptero esclavo.
xs = x(13:24);
ads = x(31:36);

% Velocidad del rotor.
ws = ones(1,4) * 10;

% Se calcula el siguiente punto a seguir.
xyzs = [xs(9), xs(11), xs(7)];
xyz_calc = calculo_anticolision(msg_xyzm, xyzs, 1);

% Llenamos los valores deseados en ceros.
xds = zeros(1, 12);

% Ganancias
alfa_s = ones(1, 12) * ganancia;

% Error
z_s = calcular_error_z_ad(xds, xs, alfa, ads);

% Control
dxs = Drone(xs, ws, alfa_s, z_s);

```

```

% Acoplamiento dinamico
%k = 100;
gamma1 = 50;
gamma2 = 100;
k = 100;
alfa = 1;

% Maestro

dxadm = zeros(1, 6);
dxadm(1) = -alfa*adm(1) + adm(2);
dxadm(2) = -gamma1*adm(1) - gamma2*adm(2) - k*(xdm(7) - xm(7));

dxadm(3) = -alfa*adm(3) + adm(4);
dxadm(4) = -gamma1*adm(3) - gamma2*adm(4) - k*(xdm(9) - xm(9));

dxadm(5) = -alfa*adm(5) + adm(6);
dxadm(6) = -gamma1*adm(5) - gamma2*adm(6) - k*(xdm(11) - xm(11));

% Esclavo
dxads = zeros(1, 6);
dxads(1) = -alfa*ads(1) + ads(2);
dxads(2) = -gamma1*ads(1) - gamma2*ads(2) - k*(xyz_calc(3) - xs(7));

dxads(3) = -alfa*ads(3) + ads(4);
dxads(4) = -gamma1*ads(3) - gamma2*ads(4) - k*(xyz_calc(1) - xs(9));

dxads(5) = -alfa*ads(5) + ads(6);
dxads(6) = -gamma1*ads(5) - gamma2*ads(6) - k*(xyz_calc(2) - xs(11));

dx = [dxm, dxs, dxadm, dxads];

dx=dx';
end

%
% Variables de estado del quadricoptero.
%
% @param x Variables de estado en una matrix de 1x12.

```

```

% @param we Velocidades del rotor en una matrix de 1x4.
% @param alfa Ganancias alfa del control backstepping en una matrix de
    1x12.
% @param z Variable de error del control backstepping en una matrix de
    1x12.
%
function dx=Drone(x, we, alfa, z)
g = -9.81; % Aceleracion de la gravedad
w1 = we(1); % Velocidad del rotor 1.
w2 = we(2); % Velocidad del rotor 2.
w3 = we(3); % Velocidad del rotor 3.
w4 = we(4); % Velocidad del rotor 4.
w = w2+w4-w1-w3;

m = .650; % Masa del cuadricoptero.
Jr = 6e-5; % Inercia del rotor.

L = .23; % Distancia desde el centro de masa hasta la base del rotor.
lx = 7.5e-3; % Inercia del cuerpo x.
ly = 7.5e-3; % Inercia del cuerpo y.
lz = 1.3e-2; % Inercia del cuerpo z.

% Control
a1 = (ly-lz)/lx;
a2 = (-Jr/lx);
a3 = (lz-lx)/ly;
a4 = (-Jr/ly);
a5 = (lx-ly)/lz;

b1 = L/lx;
b2 = L/ly;
b3 = L/lz;

u2 =
    (1/b1)*(z(1)-a1*x(4)*x(6)-a2*x(4)*w-alfa(1)*(z(2)+alfa(1)*z(1))-alfa(2)*z(2));
u3 =
    (1/b2)*(z(3)-a3*x(2)*x(6)-a4*x(2)*w-alfa(3)*(z(4)+alfa(3)*z(3))-alfa(4)*z(4));
u4 = (1/b3)*(z(5)-a5*x(2)*x(4)-alfa(5)*(z(6)+alfa(5)*z(5))-alfa(6)*z(6));
u1 = m/(cos(x(1))*cos(x(3)))*(z(7) + g -
    alfa(7)*(z(8)+alfa(7)*z(7))-alfa(8)*z(8));

```

```
ux = (m/u1)*(z(9)-alfa(9)*(z(10)+alfa(9)*z(9))-alfa(10)*z(10));
uy = (m/u1)*(z(11)-alfa(11)*(z(12)+alfa(11)*z(11))-alfa(12)*z(12));
```

```
dx = zeros(1, 12);
dx(1) = x(2);
dx(2) = x(4)*x(6)*a1 + x(4)*a2*w + b1*u2;
dx(3) = x(4);
dx(4) = x(2)*x(6)*a3 + x(2)*a4*w + b2*u3;
dx(5) = x(6);
dx(6) = x(4)*x(2)*a5 + b3*u4;
dx(7) = x(8);
dx(8) = -g + (cos(x(1))*cos(x(3))/m)*u1;
dx(9) = x(10);
dx(10) = (ux/m)*u1;
dx(11) = x(12);
dx(12) = (uy/m)*u1;
end
```

```
%
% Calcular el error z.
%
% @param xd Valores deseados en una matrix de 1x12.
% @param x Valores actuales en una matrix de 1x12.
% @param alfa Valores de ganancia en una matrix de 1x12.
% @param ad Valores del acoplamiento dinamico en una matrix de 1x3.
%
function z=calcular_error_z_ad(xd, x, alfa, ad)
z = zeros(1, 12);
z(1) = xd(1)-x(1);
z(2) = x(2)-xd(2)-alfa(1)*z(1);
z(3) = xd(3)-x(3);
z(4) = x(4)-xd(4)-alfa(3)*z(3);
z(5) = xd(5)-x(5);
z(6) = x(6)-xd(6)-alfa(5)*z(5);

z(7) = -ad(2); %%%% x
%z(7) = xd(7) - x(7); %%%% x
z(8) = x(8)-xd(8)-alfa(7)*z(7);
z(9) = -ad(4); %%%% y
%z(9) = xd(9) - x(9); %%%% y
```

```

z(10) = x(10)-alfa(9)*z(9)-xd(10);
z(11) = -ad(6); %%% z
%z(11) = xd(11) - x(11); %%% z
z(12) = x(12)-alfa(11)*z(11)-xd(12);
end

%
% Sistema anticollisiones.
%
% @param master_xyz Valores de xyz del cuadricoptero maestro.
% @param slave_xyz Valores de xyz del cuadricoptero esclavo.
% @param dist Distancia que se debe mantener entre los dos cuadricopteros.
function xyz = calculo_anticolision(master_xyz, slave_xyz, dist)
x = master_xyz(1) - slave_xyz(1);
y = master_xyz(2) - slave_xyz(2);
z = master_xyz(3) - slave_xyz(3);

distancia = sqrt(x^2 + y^2 + z^2);

if distancia == 0
distancia = 0.0001;
end

c = dist/distancia;

xyz = zeros(1, 3);

xyz(1) = master_xyz(1) - c*x;
xyz(2) = master_xyz(2) - c*y;
xyz(3) = master_xyz(3) - c*z;
end

%
% Ecuaciones parametricas para Lissajaous
%
% @param t Variable de tiempo.
%
function xyz=lemniscata(t)
a = 10;
x = (a * cos(t))/(1+(sin(t)^2));

```

```
y = (a * sin(t) * cos(t))/(1 +(sin(t)^2));  
z = 8;
```

```
xyz = zeros(1, 3);  
xyz(1) = x;  
xyz(2) = y;  
xyz(3) = z;  
end
```

```
%  
% Ecuaciones parametricas para un circulo  
%  
% @param t Variable de tiempo.  
%
```

```
function xyz=circulo(t)  
a = 10;  
x = a * cos(t);  
y = a * sin(t);  
z = 8;
```

```
xyz = zeros(1, 3);  
xyz(1) = x;  
xyz(2) = y;  
xyz(3) = z;  
end
```

```
%  
% Ecuaciones parametricas para una trisectriz  
%  
% @param t Variable de tiempo.  
%
```

```
function xyz=trisectriz(t)  
a = 10;  
x = cos(t) * (2*a*cos(t) + a);  
y = sin(t) * (2*a*cos(t) + a);  
z = 8;
```

```
xyz = zeros(1, 3);  
xyz(1) = x;
```

```

xyz(2) = y;
xyz(3) = z;
end

%
% Genera los valores de la esfera caotica.
%
% @param x Variables de caos.
%
function dx=gen_esfera_caotica(x)
b = 4.5;
c = 7;
d = -1.5;
e = 3.9;

dx = zeros(2,1);

dx(1) = b*sign(cos(x(2)))+c;
dx(2) = d*sign(sin(x(1)*sin(x(2))))+e;
end

%
% Generacion de una esfera.
%
% @param r Valor del radio.
% @param theta Valor de theta.
% @param psi Valor de psi.
% @param zoff Compensacion del valor en z.
%
function xyz=esfera(r, theta, psi, zoff)
xyz = zeros(1,3);

xyz(1) = r*sin(psi)*cos(theta);
xyz(2) = r*sin(psi)*sin(theta);
xyz(3) = r*cos(psi)+zoff;
end

%
```

```

% Generacion de un toroide.
%
% @param r Valor del radio menor.
% @param theta Valor de theta.
% @param psi Valor de psi.
% @param zoff Compensacion del valor en z.
% @param R Valor del radio mayor.
%
function xyz=toroide(r, theta, psi, zoff, R)
xyz = zeros(1,3);

xyz(1) = (R+r*cos(theta))*cos(psi);
xyz(2) = (R+r*cos(theta))*sin(psi);
xyz(3) = r*sin(theta)+zoff;
end

%
% Generacion de la trayectoria de las esfera.
%
% @param stack Valor del stack actual.
% @param sector Valor del sector actual.
% @param flags Banderas indicadoras de la esfera.
%
function [rstack, rsector, psi, theta, flag]=trayectoria_esfera(sector,
    stack, flags)
stack_size = 24;
sector_size = 30;

sectorstep = (2*pi)/sector_size;
stackstep = pi/stack_size;

if stack == -1
flags = false;
elseif stack == stack_size
flags = true;
end

if sector == sector_size
if stack < stack_size && flags == false
rstack = stack + 1;

```

```

elseif flags == true
rstack = stack - 1;
end

rsector = 0;
else
rstack = stack;
rsector = sector + 1;
end

flag = flags;

psi = stack*stackstep;
theta = sector * sectorstep;
end

%
% Generacion de la trayectoria de la media luna.
%
% @param stack Valor del stack actual.
% @param sector Valor del sector actual.
% @param flags Banderas indicadoras de la esfera.
%
function [rstack, rsector, psi, theta,
        flag]=trayectoria_media_luna(sector, stack, flags)
stack_size = 20;
sector_size = 4;

sectorstep = (2*pi)/sector_size;
stackstep = pi/stack_size;

rsector = sector;

if stack == -1
flags = false;
if sector == sector_size
rsector = 0;
else
rsector = sector + 1;
end
end

```

```
elseif stack == stack_size
flags = true;
if sector == sector_size
rsector = 0;
else
rsector = sector + 1;
end
end

if stack < stack_size && flags == false
rstack = stack + 1;
elseif flags == true
rstack = stack - 1;
end

flag = flags;

psi = stack*stackstep;
theta = sector * sectorstep;
end

% vim: ts=4 sw=4 et
```

---