

Universidad Autónoma de Baja California

Facultad de Ingeniería, Arquitectura y Diseño

Maestría y Doctorado en Ciencias e Ingeniería (MyDCI)



Formación de Cuadricópteros Utilizando un Control por Backstepping y un Acoplamiento Dinámico.

Para cubrir parcialmente los requisitos y obtener el grado de Maestro Ingeniería.

Presenta:

Johnny Rubio Pecasso

Directora de Tesis: **Dra. Rosa Martha López Gutiérrez**

Codirector de Tesis: **Dr. Adrian Arellano Delgado**

Ensenada, B.C., Junio del 2022

UNIVERSIDAD AUTÓNOMA DE BAJA CALIFORNIA

FACULTAD DE INGENIERÍA, ARQUITECTURA Y DISEÑO

MAESTRÍA Y DOCTORADO EN CIENCIAS E INGENIERÍA

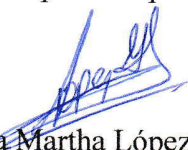
Formación de cuadricopteros utilizando un control por backstepping y un acoplamiento dinámico.

TESIS

Que para obtener el grado de maestría en ingeniería / Doctorado en Ciencias presenta:

Johnny Rubio Pecasso

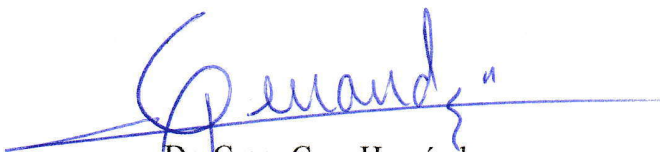
Aprobada por:



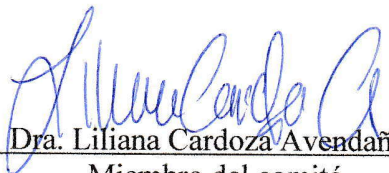
Dra. Rosa Martha López Gutiérrez
Director de tesis



Dr. Adrián Arellano Delgado
Codirector de tesis



Dr. Cesar Cruz Hernández
Miembro del comité



Dra. Liliana Cardoza Avendaño
Miembro del comité



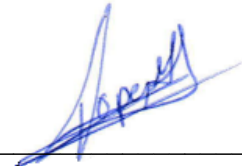
Dr. Miguel Ángel Murillo Escobar
Miembro del comité

Ensenada Baja California, México. Junio 2022.

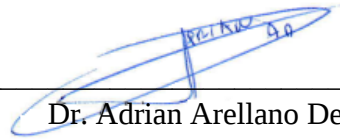
Abstract from **Johnny Rubio Pecasso** thesis, presented as a partial requirement to obtain the **Master in Science and Engineering degree** from the program *Maestría y Doctorado en Ciencias e Ingeniería (MYDCI)* from UABC. Ensenada Baja California, Mexico. June 2022.

Quadcopter Formation Using a Backstepping Control and a Dynamic Coupling.

Abstract approved by:



Dra. Rosa Martha López Gutiérrez.
Thesis Director.



Dr. Adrian Arellano Delgado.
Thesis Codirector.

In this work, a controller using backstepping control in quadcopters with Master-Slave configuration is presented through the use of a proposed dynamic coupling. By means of extensive numerical simulations, we show the advantages when comparing a system using a Backstepping control and a system using a Backstepping control with a proposed dynamic coupling. Some cases of interest are studied in order to show the advantages of using the proposed dynamic coupling scheme. These cases include different desired trajectories for the master quadcopter, in which circular, concentric and lemniscate trajectories are found; where the slave quadcopter receives the desired trajectory from the master quadcopter through dynamic coupling using a Backstepping control. The results obtained in the numerical simulations show that it is possible to achieve the formation of quadcopters using backstepping control and dynamic coupling in Master-Slave configuration.

Keywords: *Quadcopters, Backstepping Control, Dynamic Coupling, Master-Slave Configuration, Collision Avoidance, Algorithms, Chaos, Master-Intermittent.*

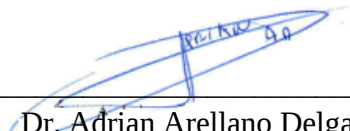
Resumen de la tesis de **Johnny Rubio Pecasso**, presentada como requisito parcial para obtener el grado de **Maestro en Ciencia e Ingeniería** del programa de *Maestría y Doctorado en Ciencias e Ingeniería (MYDCI)* de la UABC. Ensenada Baja California, México. Junio del 2022.

Formación de Cuadricópteros utilizando un control Backstepping y un Acoplamiento Dinámico.

Resumen aprobado por:



Dra. Rosa Martha López Gutiérrez.
Directora de Tesis.



Dr. Adrian Arellano Delgado.
Codirector de Tesis.

En este trabajo se presenta un controlador utilizando la técnica de backstepping en cuadricópteros con configuración Maestro-Esclavo mediante el uso de un acoplamiento dinámico propuesto. Por medio de extensas simulaciones numéricas, mostramos las ventajas al comparar un sistema utilizando un control Backstepping y con el sistema que utiliza un control Backstepping con un acoplamiento dinámico propuesto. Se estudian algunos casos de interés con el fin de mostrar las ventajas al utilizar el esquema de acoplamiento propuesto. Estos casos incluyen diferentes trayectorias deseadas para el cuadricóptero maestro, en las que se encuentran trayectoria circular, concéntrica y lemniscata; donde el cuadricóptero esclavo recibe la trayectoria deseada del cuadricóptero maestro a través de un acoplamiento dinámico usando un control Backstepping. Los resultados obtenidos en las simulaciones numéricas muestran que es posible lograr la formación de cuadricópteros utilizando control backstepping y acoplamiento dinámico en configuración Maestro-Esclavo.

Palabras claves: *Cuadricópteros, Control Backstepping, Acoplamiento Dinámico, Formación, Configuración Maestro-Esclavo, Anticolisiones, Algoritmos, Caos, Maestro-Intermitente.*

Dedicatoria

A mi madre Lenora Pecasso y mi padre Jorge Rubio, Gracias por su amor incondicional.

A mis hermanas y hermanos.

A todos los integrantes de mi familia.

A los maestros, compañeros y amigos que han aportado en mi formación.

A las personas que han sido positivas hacia mi persona; también a las que con críticas han
formado mi personalidad y carácter.

Agradecimientos

A mi familia, en especial a mi madre Lenora y mi padre Jorge, gracias por su apoyo y amor incondicional.

A mis directores Dra. Rosa Martha y Dr. Adrian, por guiarme en este proyecto de una manera inmejorable.

A mis Sinodales Dra. Liliana Cardoza, Dr. César Cruz y Dr. Miguel Murillo, por seguir de cerca mi proyecto y aportar al resultado final.

A mis compañeros de laboratorio Dr. Javier Pliego, Carlos, Michelle y Miguel por aportarme su ayuda y conocimiento.

A la Universidad Autónoma de Baja California (UABC) y su personal, por darme la oportunidad de incorporarme a su programa.

Al Centro de Investigación Científica y de Estudios Superiores de Ensenada (CICESE) y su personal, por brindarme sus instalaciones y hacerme uno más del grupo.

Al Consejo Nacional de Ciencia y Tecnología (CONACYT), por brindarme apoyo económico durante mi trabajo.

Al CONACYT a través del proyecto A1-S-31628.

Índice general

Dedicatoria	IV
Agradecimientos	V
1. Introducción	1
1.1. Motivación	2
1.2. Justificación	2
1.3. Objetivos	3
1.3.1. Objetivo general	3
1.3.2. Objetivos específicos	3
1.4. Estructura de la tesis	4
2. Preliminares	5
2.1. Estado del Arte	5
2.2. Marco Teórico	7
2.2.1. Sistemas Complejos	7
2.2.2. Control backstepping	7
2.2.3. Acoplamiento Dinámico (Dynamic Coupling)	8
2.2.4. Sincronización	9
3. Modelo matemático y análisis de control	12
3.1. Modelo matemático de un cuadricóptero	12
3.2. Introduciendo un control por backstepping	15
3.2.1. Subsistema: control de rotación.	18

3.2.2. Subsistema: control de traslación.	20
3.3. Introduciendo un acoplamiento dinámico	21
3.4. Introduciendo una Etapa de Anticolisiones	25
4. Comparación de un esquema backstepping VS un esquema backstepping con acoplamiento dinámico	26
4.1. Esquema control backstepping	26
4.2. Esquema control backstepping con acoplamiento dinámico	27
4.3. Comparación: Análisis	28
4.4. Comparación: Conclusiones	31
5. Creación de trayectorias deseadas	34
5.1. Trayectorias convencionales	34
5.1.1. Cuadricóptero individual: control de trayectoria lemniscata	34
5.1.2. Cuadricóptero individual: control de trayectoria circulo	36
5.1.3. Configuración maestro-esclavo: control de trayectoria lemniscata	37
5.1.4. Configuración maestro-esclavo: control de trayectoria circulo	37
5.1.5. Configuración maestro-esclavo: Intruso	38
5.2. Trayectorias Concéntricas	39
5.2.1. Configuración maestro-esclavo concéntrica: control de trayectoria lemniscata	39
5.2.2. Configuración maestro-esclavo concéntrica: control de trayectoria circulo	39
5.2.3. Configuración maestro-esclavo concéntrica: maestro 4 esclavos circular	40
5.2.4. Configuración maestro-esclavo concéntrica: maestro 4 esclavos lemniscata	41
5.3. Trayectorias caóticas	42
5.3.1. Configuración maestro-esclavo caótico	44
5.3.2. Configuración maestro-esclavo caótico: antifase	44
5.3.3. Configuración maestro-esclavo caótico: formación	44

5.3.4. Configuración maestro-esclavo : intruso caótico	45
6. Maestro Intermitente Caótico	46
6.1. Algoritmo	46
6.2. Simulaciones	52
7. Conclusiones	54
7.1. Aportaciones	55
7.2. Trabajo Futuro	55
Anexos	60

Índice de figuras

1.1. Cuadricóptero utilizado en tarea de inspección.	1
2.1. Puente Millenium (Londres, Inglaterra).	10
3.1. Desplazamientos y posiciones de cuadricóptero con respecto a las velocidades de los rotores.	12
3.2. Subsistemas del controlador.	15
3.3. Subsistemas del controlador: Diagrama.	16
4.1. Esquema general control backstepping.	26
4.2. Esquema backstepping con acoplamiento dinámico.	27
4.3. Control backstepping	28
4.4. Control backstepping con acoplamiento dinámico	28
4.5. Control backstepping con ganancias pequeñas.	29
4.6. Control backstepping con acoplamiento dinámico con ganancias pequeñas. . .	29
4.7. Control backstepping con ganancias grandes.	30
4.8. Control backstepping con acoplamiento dinámico con ganancias grandes. . .	30
5.1. Diagrama de flujo: maestro-esclavo.	35
5.2. Control cuadricóptero individual: lemniscata.	36
5.3. Control cuadricóptero individual: circulo.	36
5.4. Control cuadricóptero individual: lemniscata.	37
5.5. Configuración maestro-esclavo: circulo en antifase.	38
5.6. Configuración maestro-esclavo: intruso.	38

5.7. Configuración maestro-esclavo concéntrica: lemniscata.	39
5.8. Configuración maestro-esclavo concéntrica: círculo.	40
5.9. Simulación configuración maestro con 4 esclavos círculo.	40
5.10. Simulación configuración maestro con 4 esclavos lemniscata.	41
5.11. Diagrama de flujo - maestro caótico.	43
5.12. Configuración maestro-esclavo caótico.	44
5.13. Configuración maestro-esclavo caótico: antifase.	44
5.14. Configuración maestro-esclavo caótico: formado.	45
5.15. Configuración maestro-esclavo: intruso caótico.	45
6.1. Diagrama de flujo maestro intermitente.	47
6.2. Simulación maestro-intermitente 1.	49
6.3. Simulación maestro-intermitente 2.	50
6.4. Simulación maestro-intermitente 3.	50
6.5. Simulación maestro-intermitente 4.	51
6.6. Simulación maestro-intermitente 5.	51
6.7. Simulación maestro-intermitente 6.	51

Capítulo 1

Introducción

En las últimas décadas, los proyectos relacionados con el desarrollo de vehículos no tripulados (inglés UAV- Unmanned Aerial Vehicle) han sido muy importantes y estudiados, ya sea en el campo académico, científico y profesional. Dentro de las aplicaciones que se pueden encontrar uso de UAVs están, por ejemplo, la exploración de sitios de difícil acceso (en ocasiones búsqueda en desastres naturales), reconocimiento y en el uso de algoritmos de reconocimiento y el tracking de objetos. En la clasificación de UAVs se encuentran diferentes modelos y diseños de vehículos no tripulados, entre los que se encuentran los cuadricópteros (en inglés - drone).



Figura 1.1: Cuadricóptero utilizado en tarea de inspección.

1.1 Motivación

Actualmente los cuadricópteros, son dispositivos utilizados para diferentes actividades en el área de la ciencia y la tecnología. Sin embargo, por lo general las aplicaciones utilizan cuadricópteros de manera individual, para evitar sistemas más complejos. Esto último, limita las aplicaciones en las que se realizan tareas más complejas, en las que se requieran dos o más dispositivos interconectados entre sí, al intentar realizar esto se crea el problema de sistemas más complejos. Al utilizar control convencional en este tipo de sistemas, el número de parámetros y variables como la velocidad de los motores, la inercia, el peso, la fuerza de arrastre, entre otras, puede llegar a crear complicaciones en el diseño del sistema de control. Sin embargo, al aplicar control por backstepping, se busca reducir en un 50 % el número de parámetros que afectan la estabilidad del sistema, y así tener dispositivos más estables, para lograr un acoplamiento dinámico entre cuadricópteros más eficiente, lo que facilitara la interconexión entre múltiples dispositivos. Lo que conlleva a tener un arreglo de varios cuadricópteros que puedan realizar una misma tarea a la vez.

1.2 Justificación

En algunas aplicaciones es necesario tener un grupo de UAVs, para realizar las tareas de manera más productiva, por ejemplo, en la exploración de minas o en exploración en casos de desastres naturales, donde el tiempo es muy importante y la repartición de tareas entre dispositivos es importante. El estudio y relación de interacción entre robots es importante, para el desarrollo de la tecnología, debido a que en un futuro cercano este tipo de tecnologías serán las que predominarán en cualquier ámbito de la industria y vida cotidiana. Con esta investigación se busca encontrar un mejor diseño en el control y acoplamiento entre dispositivos, para tener mayor estabilidad y comunicación, y así lograr tener un algoritmo de control capaz de sincronizar dos o más dispositivos, que ayudaran a realizar tareas más especializadas y diversas. Aunque este trabajo se basa en la parte experimental y de resultados; se busca que en el futuro este trabajo sea utilizado, en el grupo de investigación para seguir desarrollando trabajos, como redes más complejas y grupos diversos de robots.

1.3 Objetivos

1.3.1 Objetivo general

Con la realización de la presente propuesta de tesis de maestría, se pretende alcanzar el siguiente objetivo general: aplicar control por backstepping y mediante un acoplamiento dinámico lograr la formación entre dos o más cuadricópteros, con el propósito de realizar diferentes trabajos colectivos.

1.3.2 Objetivos específicos

1. Obtención del modelo matemático del sistema.
2. Realizar Simulaciones en Matlab y/u otro lenguaje de programación.
3. Implementar el acoplamiento dinámico para realizar diferentes trayectorias deseadas con una configuración maestro-esclavo.

1.4 Estructura de la tesis

Este trabajo de tesis está estructurado de la manera siguiente, en el capítulo 2, se presenta el estado del arte y marco teórico que cimientan esta investigación. El capítulo 3, donde se muestra el modelo matemático de un cuadricóptero y el análisis del controlador por backstepping; además se presenta el diseño del acoplamiento dinámico y la etapa de anticolisiones. En el capítulo 4, se hace la implementación del controlador backstepping sin acoplamiento dinámico comparado con el controlador backstepping utilizando un acoplamiento dinámico. Capítulo 5, muestra la creación de trayectoria convencionales, concéntricas y caóticas, con diferentes formaciones y configuraciones maestro-esclavo. En el capítulo 6, se presenta el algoritmo que se diseñó para este trabajo llamado “maestro intermitente”. Por ultimo en el capítulo 7, se hacen las conclusiones finales del trabajo.

Capítulo 2

Preliminares

2.1 Estado del Arte

En la tabla 2.1 se muestran los resultados encontrados en diferentes bases de datos, con las palabras claves que describen a la investigación.

Esta rama de la tecnología toma ventaja de las grandes aportaciones que se ha dado dentro de la comunidad de investigadores. Los últimos trabajos muestran gran avance en el control de cuadricópteros. Una de las técnicas utilizadas para reducir el número de variables es el backstepping modificado, interconectando dos subsistemas uno de rotación angular, y un sistema de translación lineal [7]. Por lo regular se propone una estrategia de control no lineal, en condiciones en la cuales interfieran la menor cantidad de variables, donde las variables de rotación y translación se consideran cero, es cuando el cuadricóptero se encuentra flotando y el control está dado por la fuerza de arrastre (drag force) [8]. En los sistemas de control, el sliding mode control, es un método no lineal que altera la dinámica de un sistema mediante la se-

Base de datos	Resultado	Articulos Seleccionados
ScienceDirect	211	5
IEEE	9	1
Scopus	44	3
Otras	9	3
	275	12

Tabla 2.1: Revisión en Bases de Datos.

ñal discontinua, que obliga al sistema a “deslizarse” a lo largo de una sección transversal del comportamiento normal del sistema, por lo tanto, tiene una estructura de control variable. Este método ayuda ya que se pueden crear diferentes subsistemas para mejorar el control de estabilidad [9]. El trabajo se ha llevado al campo de la sincronización y formación de topología y redes de cuadricópteros, uno de los trabajos más destacados es la sincronización por medio de la comparación la dinámica de actitud y la dinámica de trayectoria; esto permite que el sistema se desacople en un inner-loop y un outer-loop, correspondiendo a la dinámica de cuadricóptero y la dinámica de error en la formación [10]. A continuación, se muestran resultados obtenidos en la revisión sistemática del estado del arte, en los cuales se realizaron la medición y búsqueda de componente, variables y tecnologías que se ha trabajado por los últimos 20 años en el ramo de la formación y control de dispositivos.

En [11] se aborda el control de un cuadróptero utilizando backstepping. El cual ayuda a reducir el número de parámetros utilizado; en comparación con un controlador convencional, como los utilizados en [8] y [9]. En este trabajo se planea utilizar un modelo de controlador similar a [12], haciendo las adecuaciones de parámetros para los cuadricópteros que se utilizarán, adecuándonos más a las especificaciones de un mini-cuadróptero como en [13]. En [14] se muestra otra implementación de un controlador no lineal, en [15] se encuentra otro trabajo similar al anterior, donde se utiliza retroalimentación por linealización. El uso de múltiples cuadricópteros utilizados en el mismo trabajo se presenta en [16], donde se puede observar el comportamiento de configuraciones líder-seguidor. El uso de simulaciones y ambientes virtuales utilizando software basado en Robot Operating System (ROS) es expuesto en [17]. Otra aproximación a la aplicación de formaciones y trayectorias entre varios cuadricópteros se muestra en [18].

Los resultados demostraron que la mayoría de los trabajos, están enfocados en el uso de cuadricópteros, en los cuales se utilizó la teoría de control lineal y control por backstepping, mostrando en su mayoría trabajos teóricos y modelos matemáticos simulados. Dentro las oportunidades que se encontraron para seguir desarrollando este trabajo están:

1. La mejora en el diseño del sistema de control y acoplamiento dinámico.
2. Utilizar topologías para la creación de formaciones entre 2 o más cuadricópteros.

2.2 Marco Teórico

2.2.1 Sistemas Complejos

Un sistema complejo, está compuesto por varias partes interconectadas o entrelazadas cuyos vínculos crean información adicional no visible ante el observador como resultado de las interacciones entre elementos. Tiene un comportamiento impredecible considerando la situación o posición que puede alcanzar en un instante preciso, sin embargo, dentro de ciertos límites, se puede conocer su comportamiento con respecto al tiempo. En teoría de control, son un arreglo de componentes físicos conectados de tal manera que el arreglo pueda comandar, dirigir o regular a sí mismo o a otro sistema. Las principales características de un sistema complejo son:

1. La emergencia del sistema.
2. Se encuentran fuera del equilibrio, no siendo posible que se auto mantengan.
3. Su capacidad para auto organizarse, centrándose en un atractor, de tal manera que a través de realimentación reducen o evitan los efectos externos, que pudiesen modificar su estructura.
4. Son sistemas adaptativos, ya que experimentan fluctuaciones ante estímulos externos, que pretendan sacarlos de su estabilidad como sistemas.

2.2.2 Control backstepping

En teoría de control, backstepping [19] es una técnica desarrollada para estabilizar sistemas dinámicos no lineales. Son sistemas contruidos por subsistemas que transmiten de un subsistema irreducible que puede ser estabilizado usando otro método de control. Este método es un sistema recursivo, ya que necesita de otro controlador para estabilizar el subsistema; el proceso termina cuando todos los controladores están estabilizados. El método de backstepping es adecuado para la “retroalimentación estricta (strict-feedback” o también llamada “triangu-

lar inferior (lower triangular)”. Un ejemplo de un sistema con retroalimentación estricta se presenta de la siguiente forma:

$$\begin{bmatrix} \xi_1 = f_1(\xi_1) + g_1(\xi_1)\xi_2 \\ \xi_2 = f_2(\xi_1, \xi_2) + g_2(\xi_1, \xi_2)\xi_3 \\ \dots \\ \xi_r = f_{r-1}(\xi_1, \xi_2 \dots \xi_{r-1}) + g_{r-1}(\xi_1, \xi_2 \dots \xi_{r-1})\xi_r \\ \xi_r = f_r(\xi_1, \xi_2 \dots \xi_r) + g_r(\xi_1, \xi_2 \dots \xi_r)\xi_r u \end{bmatrix} \quad (2.1)$$

Donde $\xi_2, \dots, \xi_r \in \mathfrak{R}, u \in \mathfrak{R}$ es la entrada y f_i, g_i para $i=1$, son las funciones conocidas. Un enfoque típico de la linealización por retroalimentación es que, en la mayoría de los casos existe la cancelación de no-linealidades útiles. El diseño de un backstepping exhibe mayor flexibilidad comparado con la linearización, ya que no requiere que el resultado de la dinámica entrada/salida sea lineal. Por otra parte, la cancelación de no linealidades puede evitarse, dando como resultado controladores menos complejos. La idea principal es utilizar algunos de las variables de estado como “controles virtuales” o “pseudocontroles”, y dependiendo de la dinámica de cada estado, diseñar leyes de control intermedias. El diseño de retroceso es un procedimiento recursivo donde se deriva una función de Lyapunov para todo el sistema. El procedimiento recursivo se puede expandir fácilmente desde el nominal caso de un sistema aumentado por un integrador. Esto también se conoce como retroceso del integrador. Basado en los principios de diseño del retroceso del integrador, el diseño de control se puede ampliar fácilmente para el caso de sistemas de retroalimentación estricta por (20).

2.2.3 Acoplamiento Dinámico (Dynamic Coupling)

Los sistemas dinámicos son sistemas en los cuales las variables estado siguen una serie de reglas temporales. Generalmente son llamados sistemas ya que están representados por un conjunto de ecuaciones (que conforman el sistema), son dinámicos porque sus parámetros varían con una variable de tiempo. Este trabajo se basa en el uso de cuadricópteros, se pretende llevar acabo la formación de dos o más cuadricópteros utilizando control por backstepping y acoplamiento dinámico.

2.2.4 Sincronización

La sincronización es la propiedad de algunos objetos en la naturaleza distinta a manifestar un ritmo informe de coexistencia, generalmente distinto a su ritmo individual y en presencia de una conexión o de un medio físico de acoplamiento. El origen proviene de las palabras *chronos* (tiempo) y *syn* (coincidencia). La sincronización se remonta cuando el investigador Christiaan Huygens observó y describió por primera vez la sincronización en el siglo XVII. Su descubrimiento fue que una pareja de péndulos de reloj que colgaban de un soporte común se lograban sincronizar, es decir, sus oscilaciones coincidían perfectamente. En el estudio que realizó Huygens, describió sus observaciones como: es bastante valor notar que cuando se suspenden dos relojes bien contruidos de dos ganchos incrustados en la misma viga de madera, el movimiento de cada péndulo en oscilaciones opuestas fue tan concordante que nunca retrocedieron en lo más mínimo uno de otro y el sonido de ellos se escuchaba simultáneamente. Además, si esta concordancia se desequilibra por alguna interferencia, se restablecía por sí mismo en un periodo de tiempo. Por largo tiempo estuve asombrado de este resultado inesperado, pero tras una cuidadosa examinación finalmente encontré que la causa de esto era debido al movimiento de la viga, aun cuando esto es difícilmente perceptible. La causa es que la oscilación de los péndulos, en proporción a sus pesos, transmite algún movimiento a los relojes. Este movimiento impuesto en la viga, necesariamente tiene el efecto de hacer que los péndulos lleguen al estado de oscilaciones exactamente contrarias si sucedió que se movieron de otra manera en un inicio y finalmente el movimiento de la viga cesa completamente [21]. De esta manera, Huygens había dado una descripción exacta, además de una excelente explicación del efecto de la sincronización mutua. En los sistemas modernos esto significa que los relojes fueron sincronizados de anti-fase debido al acoplamiento a través de la viga.

- Ejemplos de la sincronización:

La sincronización puede ser observada en la naturaleza y el día a día de las personas, a continuación, se describen algunos ejemplos de la sincronización:

1. **Sincronización en canciones de grillos.** Los grillos de pino una especie muy común, pueden sincronizar sus cantos precediendo de los cantos de sus vecinos. Esto lo convierte

en una canción larga y sincrona cuando el macho de la manada eleva sus alas delanteras y las frota. Cada chirrido entre 2 a 11 pulsos que corresponde al cierre de sus alas. Un grillo es capaz de ajustar su canción a otra como la suya rápidamente.

2. **Ritmo cardiaco.** El comportamiento de los humanos y animales se caracteriza por un “ciclo cardiaco” preciso de 24 horas de descanso y actividad. El comportamiento de este ciclo se acompaña por una oscilación diaria en la secreción hormonal y la temperatura base corporal. En las condiciones normales el ritmo cardiaco se establece por las variaciones de iluminación diaria. En experimentos en los cuales se aísla al sujeto del ciclo normal luz-obscuridad y se priva de cualquier otra señal; se observa que existe un reloj auto-sostenido que controla el ritmo. Las investigaciones demuestran que el humano tiene un marcapasos principal llamado “suprachiasmatic” del hipotálamo.
3. **Puente Millenium.** El puente Millenium (figura 2.1), ubicado en Londres, fue inaugurado el 10 de junio del 2000, por Norman Foster, con una estructura hecha en base a acero. El puente registro vibraciones en forma de “s” que llegaron alcanzar 20 cm. En desviación. La explicación a este fenómeno la dio Strogatz en su libro sync, en la que la describió como: “al caminar, creamos pequeñas fuerzas laterales con cada paso, alternativamente a la izquierda y a la derecha; estas fuerzas son pequeñas y, como normalmente no hay coordinación entre los peatones, ocurren al azar y se contrarrestan; pero al comenzar a caminar al mismo tiempo todos los asistentes, las fuerzas laterales se sumaron y se concentraron de forma sincrónica” [22].



Figura 2.1: Puente Millenium (Londres, Inglaterra).

■ **Aplicaciones de la Sincronización:**

1. **Marcapasos.** Una de las aplicaciones de la sincronización en la medicina es el marcapasos, el cual es un instrumento que envía pulsos eléctricos dentro de secciones específicas del corazón, lo cual dicta una frecuencia a los músculos cercanos. Un marcapasos es un circuito eléctrico oscilatorio (equivalente a una resistencia y un capacitor en paralelo). “El capacitor es un dispositivo que almacena voltaje, y tiene un rol relacionado a las membranas de la célula; una resistencia provee de un camino para la corriente de la célula, análogamente llamado canal de salida en la membrana” [21].
2. **Sincronización de generadores eléctricos.** El fenómeno de la sincronización en los generadores eléctricos se lleva a cabo a través de una carga común, la cual manejan estos generadores, y esta misma actúa como un factor sincronizante.

Capítulo 3

Modelo matemático y análisis de control

3.1 Modelo matemático de un cuadricóptero

Para este análisis se tiene considerado, el uso de cuadricópteros; que se caracterizan por tener 4 rotores que propulsan el vehículo. Al manipular la velocidad de cada par de rotores se puede controlar las posiciones del dispositivo. Los vehículos aéreos cuentan con 6 grados de libertad, los utilizados en el espacio (x,y,z) y 3 ángulos de inclinación en su centro de gravedad (Yaw, Pitch, Roll).

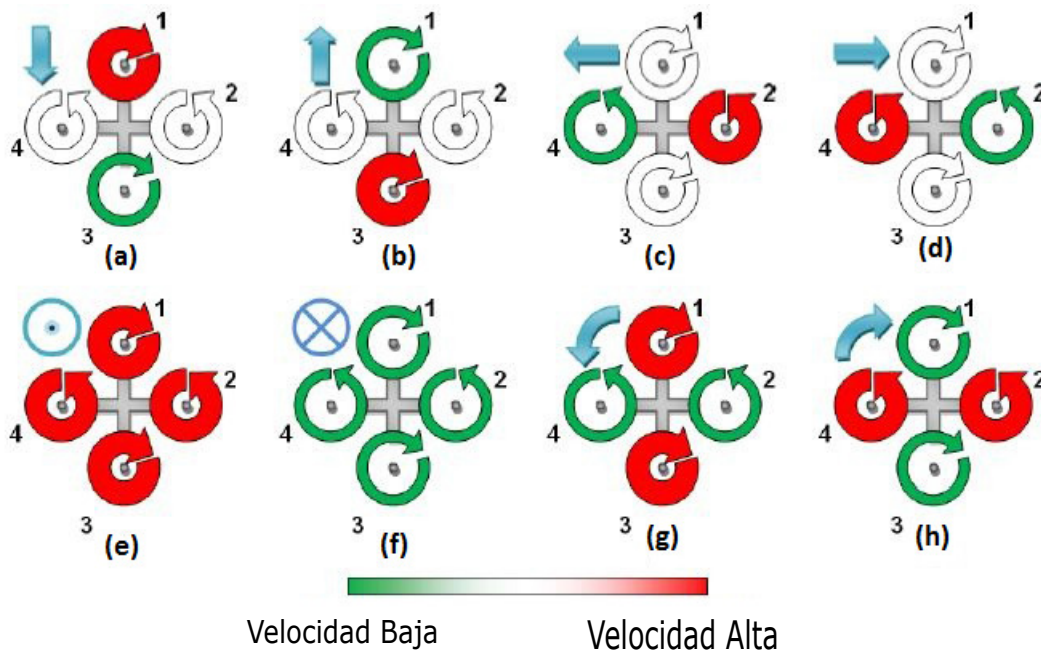


Figura 3.1: Desplazamientos y posiciones de cuadricóptero con respecto a las velocidades de los rotores.

En la figura 3.1, se muestra como se producen los movimientos de un cuadricóptero con respecto a la velocidades de los 4 rotores; **a) retroceso**, los pares 1-3 y 2-4 giran de manera inversa el rotor uno gira a máxima velocidad mientras que el rotor 3 gira a velocidad moderada; **b) avance**, tiene un comportamiento similar al inciso a) con la diferencia de que la velocidad de 1 es moderada y el rotor 3 es a máxima velocidad; **c) izquierda**, ahora las velocidades varían en el par 2-4, donde a 4 es a velocidad moderada y el 2 a velocidad máxima; **d) derecha**, como en c) pero de con un comportamiento inverso en el par 2-4; **e) elevación**, los 4 rotores giran a máxima velocidad; **f) flotando**, los 4 rotores giran a velocidad moderada; **g) giro a izquierda**, el par 1-3 a máxima velocidad y el par 2-4 a velocidad moderada; **h) giro a derecha**, el par 1-3 a velocidad moderada y el par 2-4 a máxima velocidad.

El modelo utilizado en este trabajo es uno derivado del formalismo de Newton-Euler.

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

Donde, $I \in \mathbb{R}^{3 \times 3}$, es la matriz de inercia, V es la velocidad lineal del cuerpo y ω es la velocidad angular. El modelo aproximado del cuadricóptero se puede escribir de la siguiente manera:

$$\begin{cases} \dot{\zeta} = v \\ m\dot{v} = -ge_3 + R_{e3}\left(\frac{b}{m}\right) \sum \Omega_i^2 \\ m\dot{R} = R\hat{\omega} \\ I\dot{\omega} = -\omega x I \omega - \sum J_r(\omega x e_3) \Omega_i + \tau_a \end{cases} \quad (3.2)$$

El torque aplicado en el cuerpo del vehículo sobre el eje esta descrita de la siguiente manera:

$$\tau_a = \begin{bmatrix} lb(\Omega_4^2 - \Omega_2^2) \\ lb(\Omega_3^2 - \Omega_1^2) \\ d(\Omega_2^2 + \Omega_4^2 - \Omega_1^2 - \Omega_3^2) \end{bmatrix} \quad (3.3)$$

Si se considera la inercia del motor y dando reversa a la transmisión con la inercia de negligencia, la inercia de propulsión de los pares de rotores, se tiene:

Símbolo	Definición
ζ	Posición del vector
R	Matriz de rotación
$\hat{\omega}$	Matriz del segmento simétrico
ϕ	Angulo roll
θ	Angulo pitch
ψ	Angulo yaw
Ω	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
τ_a	Torque
b	Thrust factor
d	Drag factor
l	lever

Tabla 3.1: Definiciones de símbolos.

$$J_r = J_p - J_m \quad (3.4)$$

El modelo completo se puede describir de la siguiente manera:

$$\left\{ \begin{array}{l} \ddot{\phi} = \dot{\theta} \psi \left(\frac{I_y - I_z}{I_x} \right) - \frac{J_r}{I_x} \dot{\theta} \Omega + \frac{l}{I_x} U_2 \\ \ddot{\theta} = \dot{\phi} \psi \left(\frac{I_z - I_x}{I_y} \right) + \frac{J_r}{I_y} \dot{\phi} \Omega + \frac{l}{I_y} U_3 \\ \ddot{\psi} = \dot{\phi} \dot{\theta} \left(\frac{I_x - I_y}{I_z} \right) + \frac{1}{I_z} U_4 \\ \ddot{z} = -g + (\cos \phi \cos \theta) \frac{1}{m} U_1 \\ \ddot{x} = (\cos \phi \cos \theta \cos \psi + \sin \phi \sin \psi) \frac{1}{m} U_1 \\ \ddot{y} = (\cos \phi \sin \theta \sin \psi - \sin \phi \cos \psi) \frac{1}{m} U_1 \end{array} \right. \quad (3.5)$$

Las entradas del sistema son las siguientes:

$$\begin{cases} U_1 = b(\Omega_1^2 + \Omega_2^2 + \Omega_3^2 + \Omega_4^2) \\ U_2 = b(\Omega_4^2 - \Omega_2^2) \\ U_3 = b(\Omega_3^2 - \Omega_1^2) \\ U_4 = d(\Omega_2^2 + \Omega_4^2 - \Omega_1^2 - \Omega_3^2) \\ \Omega = (\Omega_2 + \Omega_4 - \Omega_1 - \Omega_3) \end{cases} \quad (3.6)$$

3.2 Introduciendo un control por backstepping

En la fig. (3.2), se puede observar, como está dividido el sistema en dos subsistemas, uno para el control de rotación y otro para el control lineal. Las estradas U_2, U_3, U_4 están conectadas a los ángulos y sus derivadas; mientras las traslaciones lineales de penden de los ángulos obtenidos.

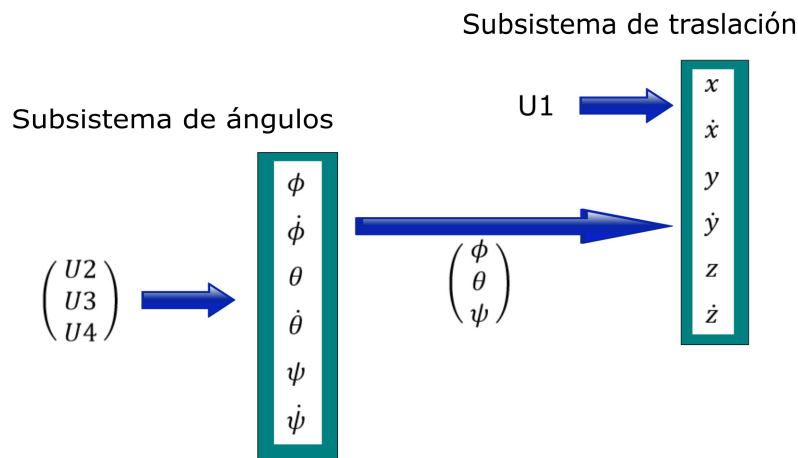


Figura 3.2: Subsistemas del controlador.

En la figura (3.3), se puede observar el esquema de control backstepping dividido en posición y rotación; donde se puede observar que las posiciones deseadas X_d, Y_d, Z_d, ψ_d ; generan los ángulos de rotación requeridos ϕ_R y θ_R . Las entradas obtenidas son retroalimentadas a los controladores de posición y rotación.

El modelo se puede reescribir, introduciendo la linearización del sistema, creando un vector de estados.

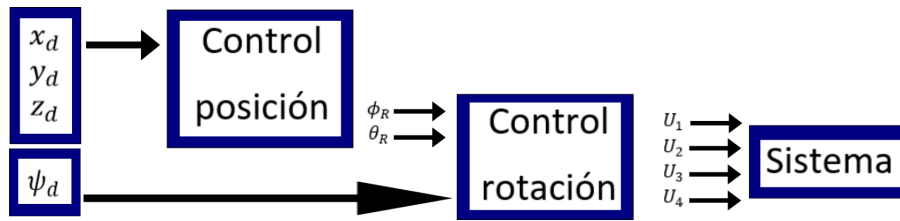


Figura 3.3: Subsistemas del controlador: Diagrama.

$$\left\{ \begin{array}{l}
 x_1 = \phi \\
 x_2 = \dot{x}_1 = \dot{\phi} \\
 x_3 = \theta \\
 x_4 = \dot{x}_3 = \dot{\theta} \\
 x_5 = \psi \\
 x_6 = \dot{x}_5 = \dot{\psi} \\
 x_7 = z \\
 x_8 = \dot{x}_7 = \dot{z} \\
 x_9 = x \\
 x_{10} = \dot{x}_9 = \dot{x} \\
 x_{11} = y \\
 x_{12} = \dot{x}_{11} = \dot{y}
 \end{array} \right. \quad (3.7)$$

Realizando un cambio de variables las ecuaciones Diferenciales quedan de la siguiente manera:

$$f(X,U) = \begin{cases} x_2 \\ x_4x_6a_1 + x_4a_2\Omega + b_1U_2 \\ x_4 \\ x_2x_6a_3 + x_2a_4\Omega + b_2U_3 \\ x_6 \\ x_4x_2a_5 + b_3U_4 \\ x_8 \\ -g + (\cos x_1 \cos x_3) \frac{1}{m} U_1 \\ x_{10} \\ U_x \frac{1}{m} U_1 \\ x_{12} \\ U_y \frac{1}{m} U_1 \end{cases} \quad (3.8)$$

Donde:

$$\begin{bmatrix} a_1 = (I_y - I_z/I_x) \\ a_2 = -J_R/I_x \\ a_3 = (I_z - I_x/I_y) \\ a_4 = J_R/I_y \\ a_5 = (I_x - I_y/I_z) \\ b_1 = l/I_x \\ b_2 = l/I_y \\ b_3 = l/I_z \\ U_x = (\cos x_1 \sin x_3 \cos x_5 + \sin x_1 \sin x_5) \\ U_y = (\cos x_1 \sin x_3 \sin x_5 - \sin x_1 \cos x_5) \end{bmatrix} \quad (3.9)$$

En la tabla (3.2), se muestran los valores que se utilizaron en las simulaciones. Para implementar el control por backstepping, se tiene que considerar la posición deseada del cuadricóp-

Símbolo	Definición	Valor
m	Masa	0.650 kg
I_x	Inercia del eje x	$7.5 \text{ e}^{-3} \text{ kg.m}^2$
I_y	Inercia del eje y	$7.5 \text{ e}^{-3} \text{ kg.m}^2$
I_z	Inercia del eje z	$1.3 \text{ e}^{-3} \text{ kg.m}^2$
b	Coeficiente de empuje	$3.13 \text{ e}^{-5} \text{ N.s}^2$
d	Coeficiente de arrastre	$7.5 \text{ e}^{-7} \text{ Nm.s}^2$
J_r	Inercia del rotor	$6 \text{ e}^{-5} \text{ kg.m}^2$
L	Tamaño de la extremidad	0.23 m

Tabla 3.2: Definiciones de símbolos y valores.

tero, con la posición obtenida por el sistema.

3.2.1 Subsistema: control de rotación.

Debido a que se están considerando subsistemas, primero se considera el control de la entrada para la rotación angular, de esto deriva la posición, utilizando el seguimiento de errores. Los errores son obtenidos por la siguiente función de Lyapunov propuesta.

$$V(z_1) = \frac{1}{2} z_1^2 \quad (3.10)$$

Por lo tanto:

$$\dot{V}(z_1) = z_1 (\dot{x}_{1d} - x_2) \quad (3.11)$$

Introduciendo una entrada de control x_2 para la estabilización, se tiene lo siguiente:

$$z_2 = x_2 - \dot{x}_{1d} - \alpha_1 z_1 \quad (3.12)$$

$$V(z_1, z_2) = \frac{1}{2} (z_1^2 + z_2^2) \quad (3.13)$$

De la misma forma se puede obtener lo siguiente:

$$\left\{ \begin{array}{l} z_3 = x_{3d} - x_3 \\ z_4 = x_4 - \dot{x}_{3d} - \alpha_3 z_3 \\ z_5 = x_{5d} - x_5 \\ z_6 = x_6 - \dot{x}_{6d} - \alpha_6 z_6 \\ z_7 = x_{7d} - x_7 \\ z_8 = x_8 - \dot{x}_{8d} - \alpha_8 z_8 \\ z_9 = x_{9d} - x_9 \\ z_{10} = x_{10} - \dot{x}_{10d} - \alpha_{10} z_{10} \\ z_{11} = x_{11d} - x_{11} \\ z_{12} = x_{12} - \dot{x}_{12d} - \alpha_{12} z_{12} \end{array} \right. \quad (3.14)$$

Derivando queda de la siguiente forma:

$$\dot{V}(z_1, z_2) = z_2(x_4 x_6 a_1 + x_4 a_2 \Omega + b_1 U_2) - z_2(\dot{x}_{1d} - \alpha_1(z_2 + \alpha_1 z_1)) - z_1 z_2 - \alpha_1 z_1^2 \quad (3.15)$$

La entrada de control U_2 esta extraída de $(x_{1,2,3d} = 0)$, satisfaciendo $V(z_1 z_2) > 0$.

$$U_2 = \frac{1}{b_1}(z_1 - x_4 x_6 a_1 - x_4 a_2 \Omega - \alpha_1(z_2 + \alpha_1 z_1) - \alpha_2 z_2) \quad (3.16)$$

El termino $\alpha_2 z_2$, cuando $\alpha_2 > 0$ es agregado para estabilizar z_1 .

$$U_3 = \frac{1}{b_2}(z_3 - x_2 x_6 a_3 - x_2 a_4 \Omega - \alpha_3(z_4 + \alpha_3 z_3) - \alpha_3 z_3) \quad (3.17)$$

$$U_4 = \frac{1}{b_3}(z_5 - x_2 x_4 a_5 - \alpha_5(z_6 + \alpha_5 z_5) - \alpha_6 z_6) \quad (3.18)$$

3.2.2 Subsistema: control de traslación.

Ahora, al diseñar el subsistema de traslación, el control de U_1 se obtiene utilizando el mismo enfoque que antes.

$$U_1 = \frac{m}{\cos x_1 \cos x_3} (z_7 + g - \alpha_7(z_8 + \alpha_7 z_7) - \alpha_8 z_8) \quad (3.19)$$

El control del movimiento en x, y se representa de la siguiente forma:

$$U_x = \frac{m}{U_1} (z_9 - \alpha_9(z_{10} + \alpha_9 z_9) - \alpha_{10} z_{10}) \quad (3.20)$$

$$U_y = \frac{m}{U_1} (z_{11} - \alpha_{11}(z_{12} + \alpha_{11} z_{11}) - \alpha_{12} z_{12}) \quad (3.21)$$

Para realizar el cálculo de ϕ_R y θ_R , se obtienen de las posiciones deseadas de la siguiente manera:

$$U_z = m * (z_7 - \alpha_7 * (z_8 + \alpha_7) - \alpha_8 * z_8) \quad (3.22)$$

Ahora bien, utilizando las posiciones X, Y se obtiene el siguiente sistema de ecuaciones:

$$\begin{bmatrix} U_x \\ U_y \end{bmatrix} = (U_z + mg) \begin{bmatrix} \cos \psi_d & \sin \psi_d / \cos \theta_d \\ \sin \psi_d & -\cos \psi_d / \cos \theta_d \end{bmatrix} \begin{bmatrix} \tan(\theta_d) \\ \tan(\phi_d) \end{bmatrix} \quad (3.23)$$

Encontrando las soluciones del sistema, queda de la siguiente manera:

$$\tan(\theta_R) = \frac{1}{U_z + mg} (\cos \psi_d U_x + \sin \psi_d U_y) \quad (3.24)$$

$$\tan(\phi_R) = \frac{\cos \theta_R}{U_z + mg} (\sin \psi_d U_x - \cos \psi_d U_y) \quad (3.25)$$

Realizando el despeje los algunos requeridos quedan de la siguiente forma:

$$\theta_R = \tan^{-1} \left(\frac{1}{U_z + mg} (Cos \psi_d U_x + Sen \psi_d U_y) \right) \quad (3.26)$$

$$\phi_R = \tan^{-1} \left(\frac{Cos \theta_R}{U_z + mg} (Sen \psi_d U_x - Cos \psi_d U_y) \right) \quad (3.27)$$

Respetando el cambio de variable de la ecuación (3.7), las ecuaciones (3.26) y (3.27); quedan de la siguiente forma:

$$x_1 = \tan^{-1} \left(\frac{1}{U_z + mg} (Cos(x_5) U_x + Sen(x_5) U_y) \right) \quad (3.28)$$

$$x_3 = \tan^{-1} \left(\frac{Cos(x_1)}{U_z + mg} (Sen(x_5) U_x - Cos(x_5) U_y) \right) \quad (3.29)$$

3.3 Introduciendo un acoplamiento dinámico

Los sistemas dinámicos son sistemas en los que las variables de estado siguen una serie de reglas temporales. Generalmente llamados sistemas, ya que están representados por un conjunto de ecuaciones; son dinámicos porque sus parámetros varían con una variable de tiempo. El acoplamiento dinámico propuesto, está inspirado en el experimento de Huygens. Las siguientes ecuaciones representan el acoplamiento dinámico propuesto en el cuadricóptero maestro.

$$h_\phi = \begin{cases} \dot{h}_{1\phi} = -h_{1\phi} + h_{2\phi} \\ \dot{h}_{2\phi} = -\gamma_1 h_{1\phi} - \gamma_2 h_{2\phi} - k(\phi_R - x_1) \end{cases} \quad (3.30)$$

$$h_\theta = \begin{cases} \dot{h}_{1\theta} = -h_{1\theta} + h_{2\theta} \\ \dot{h}_{2\theta} = -\gamma_1 h_{1\theta} - \gamma_2 h_{2\theta} - k(\theta_R - x_3) \end{cases} \quad (3.31)$$

$$h_\psi = \begin{cases} \dot{h}_{1\psi} = -h_{1\psi} + h_{2\psi} \\ \dot{h}_{2\psi} = -\gamma_1 h_{1\psi} - \gamma_2 h_{2\psi} - k(\psi_d - x_5) \end{cases} \quad (3.32)$$

$$h_z = \begin{cases} \dot{h}_{1z} = -h_{1z} + h_{2z} \\ \dot{h}_{2z} = -\gamma_1 h_{1z} - \gamma_2 h_{2z} - k(z_d - x_7) \end{cases} \quad (3.33)$$

$$h_x = \begin{cases} \dot{h}_{1x} = -h_{1x} + h_{2x} \\ \dot{h}_{2x} = -\gamma_1 h_{1x} - \gamma_2 h_{2x} - k(x_d - x_9) \end{cases} \quad (3.34)$$

$$h_y = \begin{cases} \dot{h}_{1y} = -h_{1y} + h_{2y} \\ \dot{h}_{2y} = -\gamma_1 h_{1y} - \gamma_2 h_{2y} - k(y_d - x_{11}) \end{cases} \quad (3.35)$$

Donde γ_1, γ_2 son constantes positivas, k es la fuerza de acoplamiento, $\Delta_x, \Delta_y, \Delta_z$, son desplazamientos de (x, y, z) , respectivamente. Estos desplazamientos constantes se introducen para separar los cuadricópteros en el plano (x, y, z) . Usando el acoplamiento dinámico, los errores para el cuadricóptero maestro son los siguientes:

$$Z_i = \begin{cases} z_1 = -\dot{h}_{2\phi} \\ z_2 = x_2 - a_1 z_1 \\ z_3 = -\dot{h}_{2\theta} \\ z_4 = x_4 - a_4 z_3 \\ z_5 = -\dot{h}_{2\psi} \\ z_6 = x_6 - a_6 z_5 \\ z_7 = -\dot{h}_{2z} \\ z_8 = x_8 - a_8 z_7 \\ z_9 = -\dot{h}_{2x} \\ z_{10} = x_{10} - a_{10} z_9 \\ z_{11} = -\dot{h}_{2y} \\ z_{12} = x_{12} - a_{12} z_{11} \end{cases} \quad (3.36)$$

Para el sistema esclavo, las ecuaciones de estado son:

$$f(X_s, U_s) = \begin{cases} x_{2s} \\ x_{4s}x_{6s}a_{1s} + x_{4s}a_{2s}\Omega + b_1U_{2s} \\ x_{4s} \\ x_{2s}x_{6s}a_{3s} + x_{2s}a_{4s}\Omega + b_2U_{3s} \\ x_{6s} \\ x_{4s}x_{2s}a_{5s} + b_3U_{4s} \\ x_{8s} \\ -g + (\cos x_{1s}\cos x_{3s})\frac{1}{m}U_{1s} \\ x_{10s} \\ U_{xs}\frac{1}{m}U_{1s} \\ x_{12s} \\ U_{ys}\frac{1}{m}U_{1s} \end{cases} \quad (3.37)$$

La implementación del acoplamiento dinámico en el sistema esclavo se describe a través de lo siguiente:

$$h_{\phi s} = \begin{cases} \dot{h}_{1\phi s} = -h_{1\phi s} + h_{2\phi s} \\ \dot{h}_{2\phi s} = -\gamma_1 h_{1\phi s} - \gamma_2 h_{2\phi s} - k(\beta_1 x_1 - x_{1s} - \Delta_{\phi s}) \end{cases} \quad (3.38)$$

$$h_{\theta s} = \begin{cases} \dot{h}_{1\theta s} = -h_{1\theta s} + h_{2\theta s} \\ \dot{h}_{2\theta s} = -\gamma_1 h_{1\theta s} - \gamma_2 h_{2\theta s} - k(\beta_2 x_3 - x_{3s} - \Delta_{\theta s}) \end{cases} \quad (3.39)$$

$$h_{\psi s} = \begin{cases} \dot{h}_{1\psi s} = -h_{1\psi s} + h_{2\psi s} \\ \dot{h}_{2\psi s} = -\gamma_1 h_{1\psi s} - \gamma_2 h_{2\psi s} - k(\beta_3 x_5 - x_{5s} - \Delta_{\psi s}) \end{cases} \quad (3.40)$$

$$h_{zs} = \begin{cases} \dot{h}_{1zs} = -h_{1zs} + h_{2zs} \\ \dot{h}_{2zs} = -\gamma_1 h_{1zs} - \gamma_2 h_{2zs} - k(\beta_4 x_7 - x_{7s} - \Delta_{zs}) \end{cases} \quad (3.41)$$

$$h_{xs} = \begin{cases} \dot{h}_{1xs} = -h_{1xs} + h_{2xs} \\ \dot{h}_{2xs} = -\gamma_1 h_{1xs} - \gamma_2 h_{2xs} - k(\beta_5 x_9 - x_{9s} - \Delta_{xs}) \end{cases} \quad (3.42)$$

$$h_{ys} = \begin{cases} \dot{h}_{1ys} = -h_{1ys} + h_{2ys} \\ \dot{h}_{2ys} = -\gamma_1 h_{1ys} - \gamma_2 h_{2ys} - k(\beta_6 x_{11} - x_{11s} - \Delta_{ys}) \end{cases} \quad (3.43)$$

Donde $\Delta_x, \Delta_y, \Delta_z$, son desplazamientos de (x, y, z). Las constantes de desplazamiento son introducidas individualmente en el cuadricoptero. Nótese que se introduce una variable de signo β_i , con $i=1,2,\dots,6$ para generar sincronización en antifase. Los errores del cuadricóptero esclavo son los siguientes:

$$Z_{is} = \begin{cases} z_{1s} = -\dot{h}_{2\phi s} \\ z_{2s} = x_{2s} - a_1 z_{1s} \\ z_{3s} = -\dot{h}_{2\theta s} \\ z_{4s} = x_{4s} - a_4 z_{3s} \\ z_{5s} = -\dot{h}_{2\psi s} \\ z_{6s} = x_{6s} - a_6 z_{5s} \\ z_{7s} = -\dot{h}_{2z s} \\ z_{8s} = x_{8s} - a_8 z_{7s} \\ z_{9s} = -\dot{h}_{2xs} \\ z_{10s} = x_{10s} - a_{10} z_{9s} \\ z_{11s} = -\dot{h}_{2ys} \\ z_{12s} = x_{12s} - a_{12} z_{11s} \end{cases} \quad (3.44)$$

3.4 Introduciendo una Etapa de Anticolisiones

El esquema de acoplamiento propuesto, si funciona correctamente, sin embargo, no cuenta con un sistema que pueda evitar choques entre los cuadricópteros, por lo que se recurre a modificar el acoplamiento dinámico, basado en un modelo similar [34], adaptado para este trabajo. La etapa de anticolisión implementada en este trabajo es el siguiente:

$$\dot{h}_{2z} = -\gamma_1 h_{1z} - \gamma_2 h_{2z} - k(\beta_4 x_7 - x_{7s} - \Delta_z - r_{ms} \text{sign}(x_7 - x_{7s})) \quad (3.45)$$

$$\dot{h}_{2x} = -\gamma_1 h_{1x} - \gamma_2 h_{2x} - k(\beta_5 x_9 - x_{9s} - \Delta_x - r_{ms} \text{sign}(x_9 - x_{9s})) \quad (3.46)$$

$$\dot{h}_{2y} = -\gamma_1 h_{1y} - \gamma_2 h_{2y} - k(\beta_6 x_{11} - x_{11s} - \Delta_y - r_{ms} \text{sign}(x_{11} - x_{11s})) \quad (3.47)$$

Donde, r_{ms} es el coeficiente anticolisión y m es el umbral de colisión, que puede tomar los siguientes valores:

$$r_{ms} = \begin{cases} 0, si \parallel \vec{r_{ms}} \parallel > m, \\ 0, si \parallel \vec{r_{ms}} \parallel \leq m. \end{cases} \quad (3.48)$$

En la norma euclidiana formada por la posición de cada uno de los ejes se obtiene la magnitud, es decir:

$$\parallel \vec{L_{ms}} \parallel = \sqrt{(x_9 - x_{9s})^2 + (x_{11} - x_{11s})^2 + (x_7 - x_{7s})^2} \quad (3.49)$$

Donde $\text{sign}(L_{ms})$, es la función de signo representada como:

$$\text{sign}(L_{ms}) = \begin{cases} 1, si & L_{ms} > 0, \\ 0, si & L_{ms} = 0, \\ -1, si & L_{ms} < 0. \end{cases} \quad (3.50)$$

Capítulo 4

Comparación de un esquema backstepping VS un esquema backstepping con acoplamiento dinámico

4.1 Esquema control backstepping

En la figura 4.1, se muestra el diagrama del esquema general del control backstepping, con configuración maestro-esclavo, donde U_m, X_m, U_s , representan las entradas de control del cuadricóptero maestro, las variables de estados del sistema maestro y las entradas de control del cuadricóptero Esclavo, respectivamente.

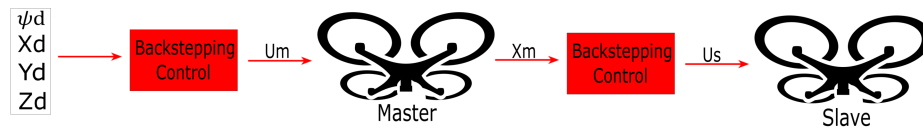


Figura 4.1: Esquema general control backstepping.

4.2 Esquema control backstepping con acoplamiento dinámico

En la figura 4.2, se muestra el diagrama del esquema general del control backstepping con acoplamiento dinámico, con configuración maestro-esclavo, donde $U_m, X_m, U_s, X_s, h_{2m}, h_{2s}$, representan las entradas de control del cuadricóptero maestro, las variables de estado del sistema maestro, las entradas de control del cuadricóptero Esclavo, las variables de estado del sistema Esclavo, las variables de estado del acoplamiento dinámico maestro y las variables de estado del acoplamiento dinámico Esclavo, respectivamente.

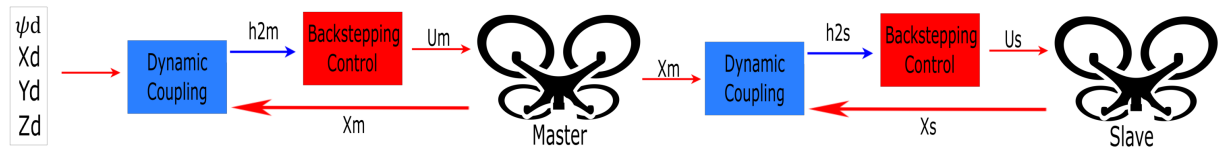


Figura 4.2: Esquema backstepping con acoplamiento dinámico.

4.3 Comparación: Análisis

El siguiente análisis, muestra el comportamiento que existe en un sistema maestro utilizando control por backstepping comparado con un sistema maestro utilizando un acoplamiento dinámico.

Para el primer experimento figura 4.3, se utilizaron ganancias iguales en ambos sistemas, los resultados obtenidos son los siguientes:

- En la figura 4.3, la Ganancia (α) = 200; el tiempo de computo fue de 3.125871 Segundos.

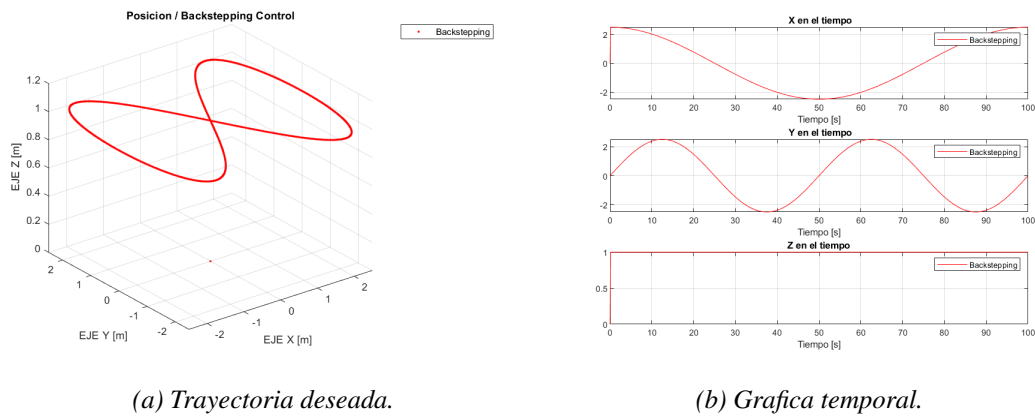


Figura 4.3: Control backstepping

- En la figura 4.4, la Ganancia (α) = 200 y la Fuerza de acoplamiento dinámico (k) = 200; el tiempo de computo fue de 1.287702 Segundos.

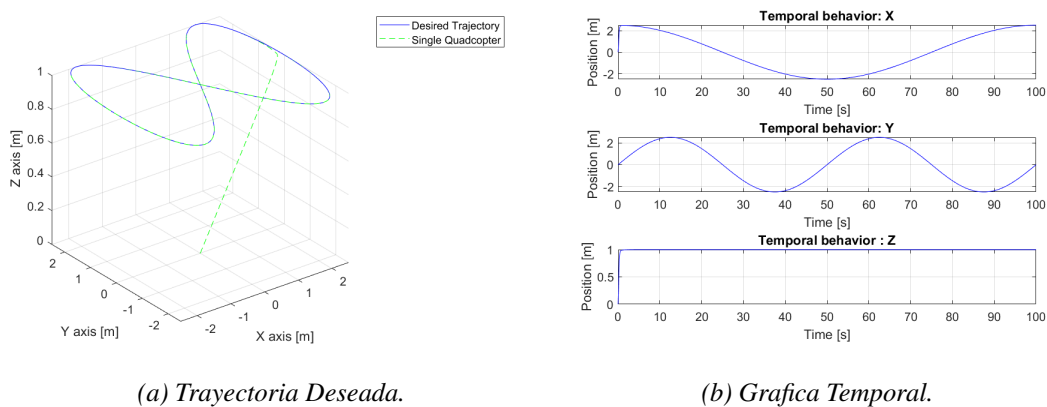


Figura 4.4: Control backstepping con acoplamiento dinámico

En este análisis donde se tienen ganancias iguales en ambos sistemas se puede concluir que el tiempo de computo es 250 % más rápido en el sistema dinámico.

Para analizar cómo se comportan los sistemas con ganancias pequeñas o mínimas, se analizó el sistema con los siguientes parámetros (figura 4.5) y (4.6):

- En la figura 4.5, la Ganancia (α) = 1; el tiempo de computo fue de 0.582050 Segundos.

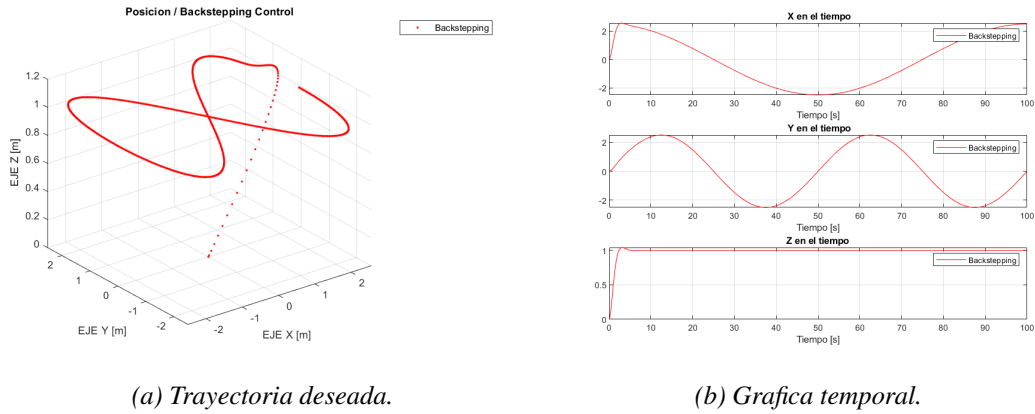


Figura 4.5: Control backstepping con ganancias pequeñas.

- En la figura 4.6, la Ganancia (α) = 1 y la Fuerza de acoplamiento dinámico (k) = 1; el tiempo de computo fue de 0.355879 Segundos.

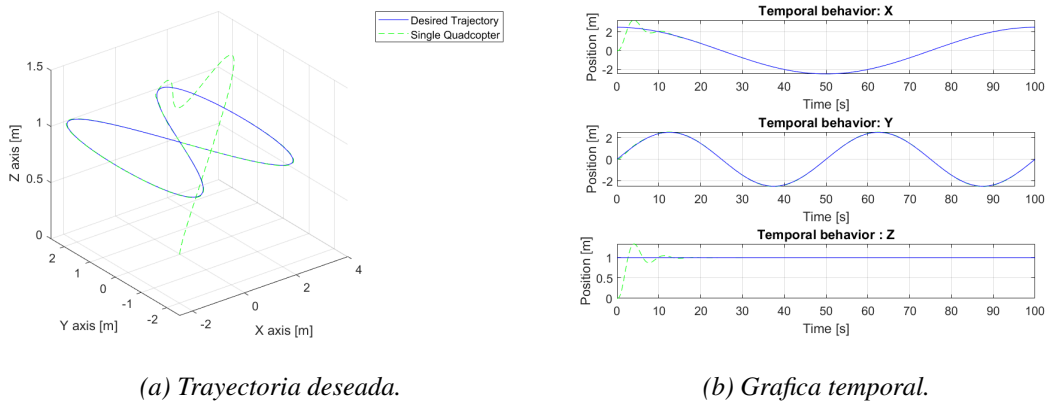


Figura 4.6: Control backstepping con acoplamiento dinámico con ganancias pequeñas.

Con ganancias pequeñas se observó que el comportamiento del transitorio en el sistema dinámico se vio más afectado que en el control por backstepping; en cuanto a tiempo de computo el sistema dinámico sigue siendo mejor, pero ahora con alrededor del 75 % comparado con backstepping.

- En la figura 4.7, la Ganancia (α) = 2000; el tiempo de computo fue de 23.215402 Segundos.

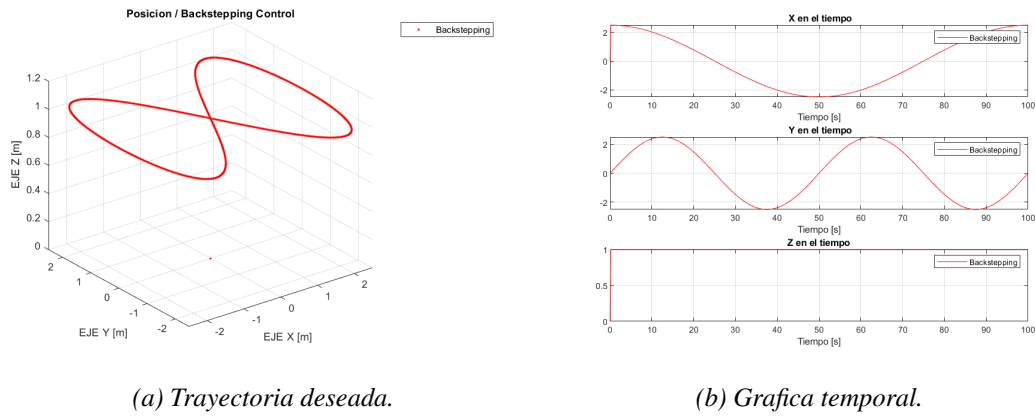


Figura 4.7: Control backstepping con ganancias grandes.

- En la figura 4.8, la Ganancia (α) = 2000 y la Fuerza de acoplamiento dinámico (k) = 2000; el tiempo de computo fue de 8.319175 Segundos.

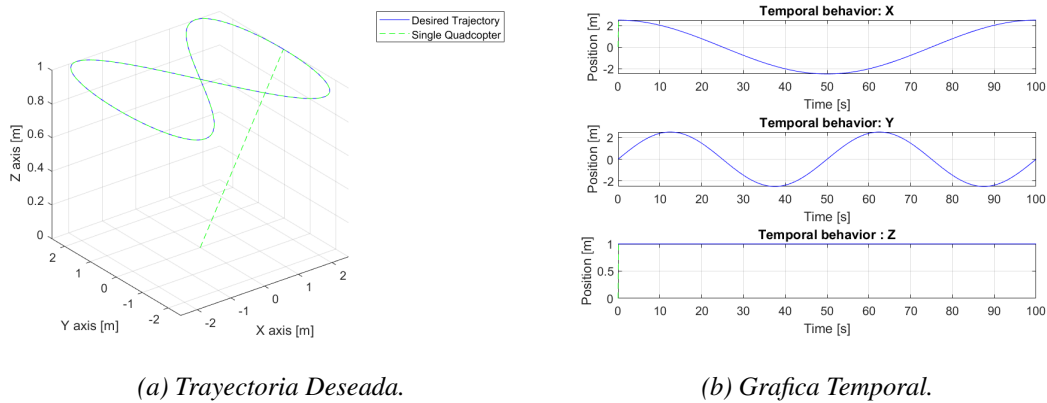


Figura 4.8: Control backstepping con acoplamiento dinámico con ganancias grandes.

En conclusión, con ganancias relativamente grandes (figura 4.7) y (figura 4.8) se puede apreciar que el sistema tiene un comportamiento, tanto transitorio como estacionario, se comportan de manera similar.

	Backstepping sin acoplamiento dinámico	Backstepping con acoplamiento dinámico
Transitorio	En general tiene un mejor comportamiento con cualquier ganancia.	Tiene un mejor comportamiento cuando las ganancias son relativamente medianas o grandes.
Estacionario	El estacionario tiene comportamiento estable con cualquier ganancia.	El estacionario tiene comportamiento estable con cualquier ganancia.
Tiempo	En general el tiempo de computo es similar y exponencial con respecto a la ganancia.	El tiempo de computo es similar con cualquier ganancia. Pero mayor al obtenido con el método por backstepping.

Tabla 4.1: Comparaciones entre ambos esquemas.

Backstepping	U1	U2	U3	U4	Tiempo (s)	Estabilidad
$\alpha = 1$	-5.0765	0.0114	0.0114	0.0197	0.5020	X = 115
$\alpha = 10$	59.2735	0.5748	0.5748	0.9964	0.6319	X = 115
$\alpha = 100$	6.4943e+03	56.9186	56.9186	98.6589	1.2234	X = 115
$\alpha = 1000$	6.4999e+05	5.6913e+03	5.6913e+03	9.8649e+03	5.4612	X = 115
$\alpha = 10000$	6.5000e+07	5.6913e+05	5.6913e+05	9.8649e+05	44.4554	X = 130

Tabla 4.2: Comportamiento backstepping.

4.4 Comparación: Conclusiones

Se puede concluir que con ganancias relativamente grandes se puede apreciar que el sistema tiene un comportamiento, tanto transitorio como estacionario, se comportan de manera similar, a continuación, en la tabla 4.1 se resumen ventajas y desventajas de ambos sistemas.

En cuanto al tiempo de complejidad en el que se resuelven ambos algoritmos, es sistema dinámico compromete el tiempo, ya que se resuelven más sistemas de Ecuaciones Diferenciales; sin embargo, la mayor ventaja son las magnitudes en las 4 entradas (U1, U2, U3, U4), en las cuales se mostró que el sistema dinámico presento entradas con magnitudes más pequeñas, lo que se traduce a picos menos pronunciados, los que ayudaran a tener un sistema físico menos comprometido. En la siguientes tablas (4.2) y (4.3) se muestra como las ganancias de cada sistema afecta al tiempo de complejidad y la estabilidad de las entradas.

En conclusión, se establece que, aunque el tiempo de complejidad es mínimamente superior en el sistema dinámico, se puede tener una mejor ventaja en cuestión de las magnitudes

Acoplamiento Dinámico	U1	U2	U3	U4	Tiempo (s)	Estabilidad
$\alpha = 1$						
$k = 1$	-6.1094	0.0025	0.0023	0.0041	1.4411	X = 240
$k = 10$	-5.5034	0.0077	0.0076	0.0133	1.6835	X = 200
$k = 100$	-5.1409	0.0108	0.0108	0.0188	2.1467	X = 700
$k = 1000$	-5.0809	0.0113	0.0113	0.0197	7.9077	No estable
$k = 10000$	-5.0707	0.0114	0.0114	0.0198	71.4600	No estable
$\alpha = 10$						
$k = 1$	-3.6642	0.0241	0.0234	0.0412	1.5738	X = 500
$k = 10$	7.4388	0.1247	0.1172	0.2096	1.6826	X = 155
$k = 100$	38.2365	0.3890	0.3892	0.6771	2.3177	X = 200
$k = 1000$	56.1105	0.5472	0.5471	0.9484	7.7097	X = 950
$k = 10000$	58.7184	0.5700	0.5700	0.9879	70.7559	No estable
$\alpha = 100$						
$k = 1$	25.7273	0.2817	0.2805	0.4872	2.8855	No estable
$k = 10$	262.6770	2.3925	2.3185	4.0829	2.7471	X = 540
$k = 100$	1.3674e+03	12.3924	11.6545	20.8407	3.2204	X = 150
$k = 1000$	4.4166e+03	38.8620	38.5903	67.1253	8.2228	X = 200
$k = 10000$	6.1799e+03	54.1696	54.1629	93.8882	70.9314	X = 375
$\alpha = 1000$						
$k = 1$	321.6019	2.8724	2.8711	4.9777	14.5783	No estable
$k = 10$	3.2037e+03	28.1649	28.0494	48.7191	14.7687	No estable
$k = 100$	2.6900e+04	239.2567	231.8593	408.3006	14.5107	X = 700
$k = 1000$	1.3741e+05	1.2396e+03	1.1658e+03	2.0847e+03	17.6555	X = 155
$k = 10000$	4.4229e+05	3.8863e+03	3.8591e+03	6.7126e+03	67.6470	X = 200
$\alpha = 10000$						
$k = 1$	3.2806e+03	28.7806	28.7793	49.8852	139.5636	No estable
$k = 10$	3.2791e+04	287.2350	287.1092	497.7650	139.8522	No estable
$k = 100$	3.2101e+05	2.8165e+03	2.8049e+03	4.8719e+03	138.6082	No estable
$k = 1000$	2.6907e+06	2.3926e+04	2.3186e+04	4.0831e+04	145.9142	X = 700
$k = 10000$	1.3742e+07	1.2396e+05	1.1658e+05	2.0847e+05	155.1919	X = 150

Tabla 4.3: Comportamiento backstepping con acoplamiento dinámico.

de las entradas, lo que se traduce a tener mayor seguridad en el equipo físico, es decir el sistema dinámico funciona como un filtro en el cual pasan por él, todos los picos antes de llegar al controlador del cuadricóptero, siendo este un sistema intermediario.

Capítulo 5

Creación de trayectorias deseadas

Todas las simulaciones y análisis se implementan en Matlab 2019, utilizando la función ODE 45 para resolver ecuaciones deferentes. Se muestran diferentes configuraciones de trayectorias deseadas para demostrar el correcto desempeño del diseño del controlador para este trabajo. En la figura 5.1 se presenta el diagrama de bloques del esquema de acoplamiento propuesto.

5.1 Trayectorias convencionales

5.1.1 Cuadricóptero individual: control de trayectoria lemniscata

La trayectoria de la lemniscata está representada por las siguientes ecuaciones:

$$lemniscata = \begin{cases} Xd = 2.5\cos(0.2t) \\ Yd = 2.5\sin(0.4t) \\ Zd > 0 \end{cases} \quad (5.1)$$

Donde X_d, Y_d, Z_d , representan la posición deseada en cada eje del plano, y el ángulo de rotación yaw deseado es $\psi_d = 10$. En la figura 5.2, se muestran los gráficos de simulación que describen una trayectoria de lemniscata, el comportamiento temporal de las posiciones (X, Y,

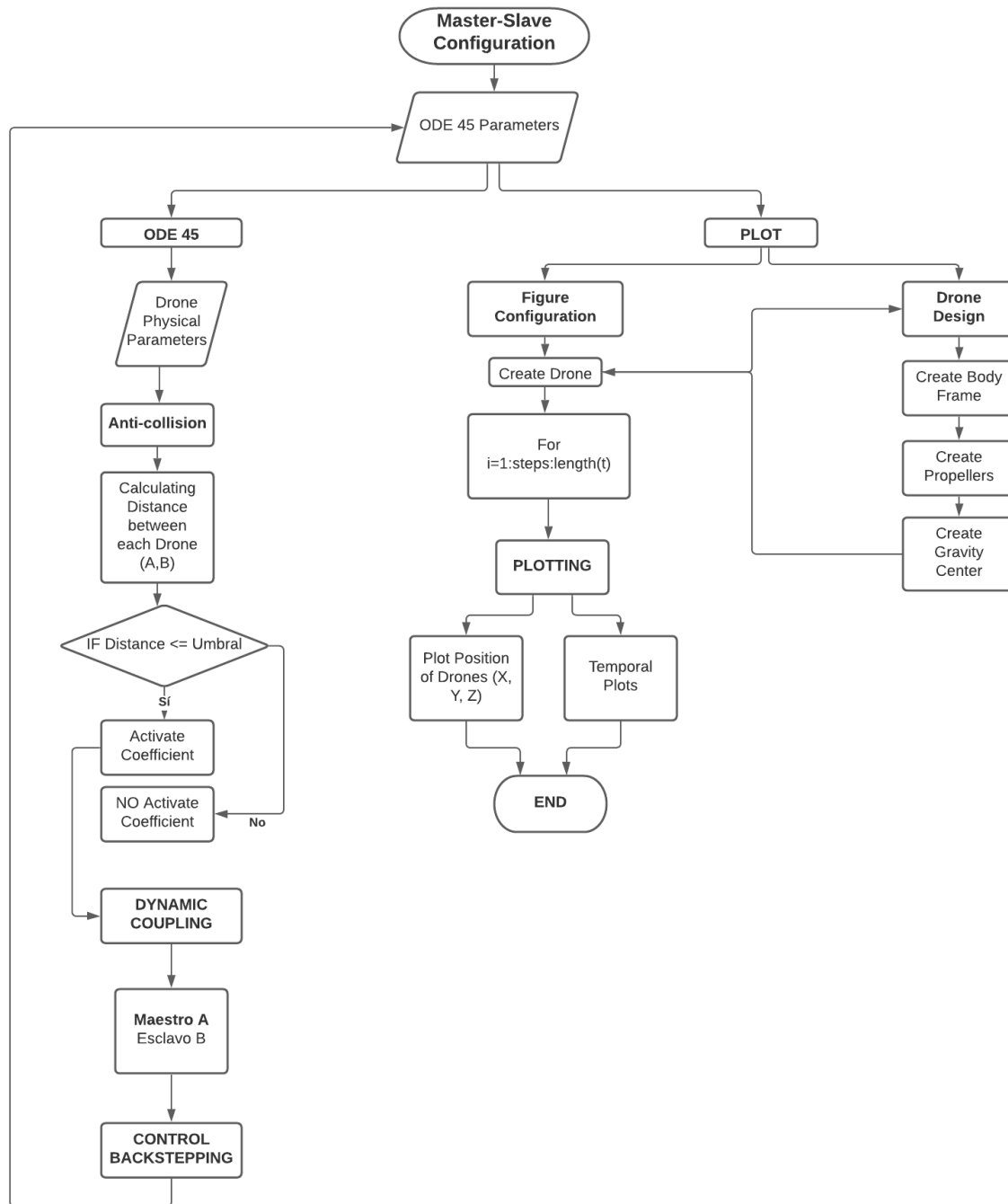


Figura 5.1: Diagrama de flujo: maestro-esclavo.

Z) y el comportamiento de estabilidad de los ángulos de rotación, respectivamente, donde se demuestra que la trayectoria deseada de seguimiento del cuadricóptero está teniendo un buen comportamiento.

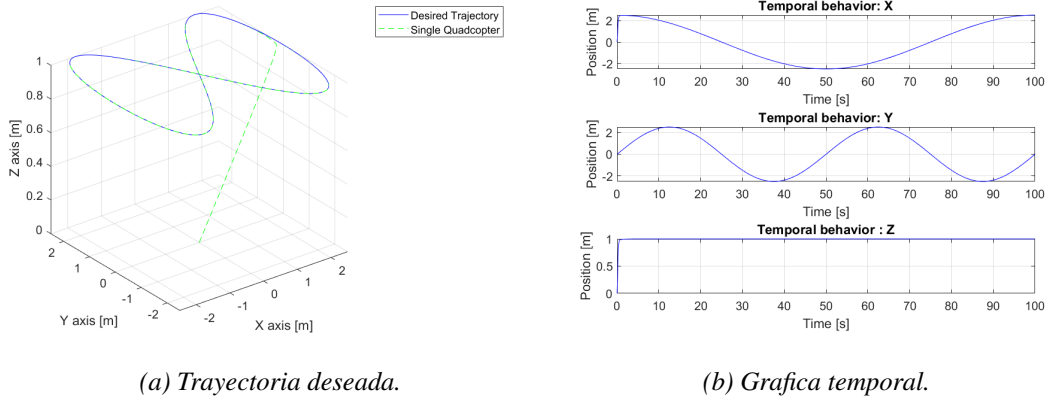


Figura 5.2: Control cuadricóptero individual: lemniscata.

5.1.2 Cuadricóptero individual: control de trayectoria circular

Las siguientes ecuaciones representan una trayectoria circular de la siguiente manera:

$$circulo = \begin{cases} Xd = 3\sin\left(\left(\frac{2\Pi}{p}\right) * t\right) \\ Yd = 3\cos\left(\left(\frac{2\Pi}{p}\right) * t\right) \\ Zd > 0 \end{cases} \quad (5.2)$$

El ángulo yaw de rotación deseado es $\psi_d = 10$.

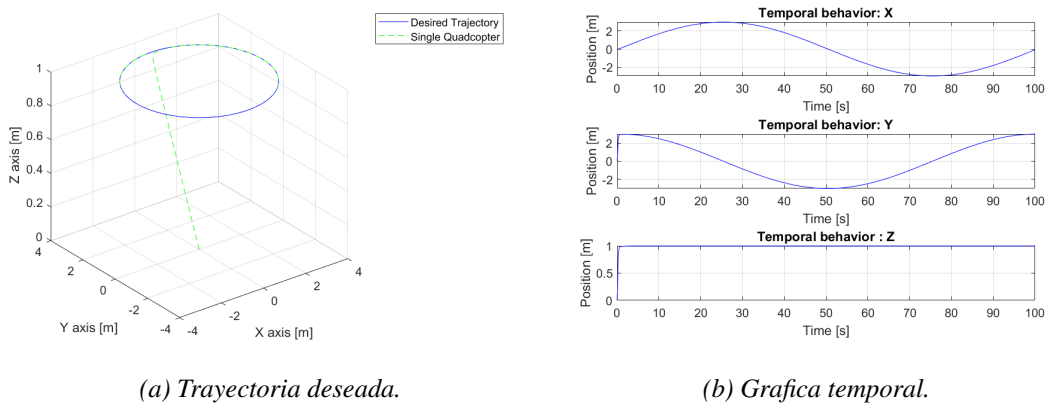


Figura 5.3: Control cuadricóptero individual: círculo.

En la figura 5.3, se representan la trayectoria de deseo de seguimiento circular de un solo cuadricóptero, la posición temporal y los ángulos de rotación, respectivamente, mostrando un buen desempeño del sistema.

5.1.3 Configuración maestro-esclavo: control de trayectoria lemniscata

La siguiente simulación representa el comportamiento de la Configuración maestro-esclavo siguiendo una trayectoria de lemniscata, donde está funcionando el anticollisiones, con valores $m = 1, r_{ms} = 0.1$, del coeficiente de anticollisiones y el umbral de anticollisiones respectivamente, estos mismos los valores utilizados en la anticollisiones que se utiliza en las simulaciones posteriores.

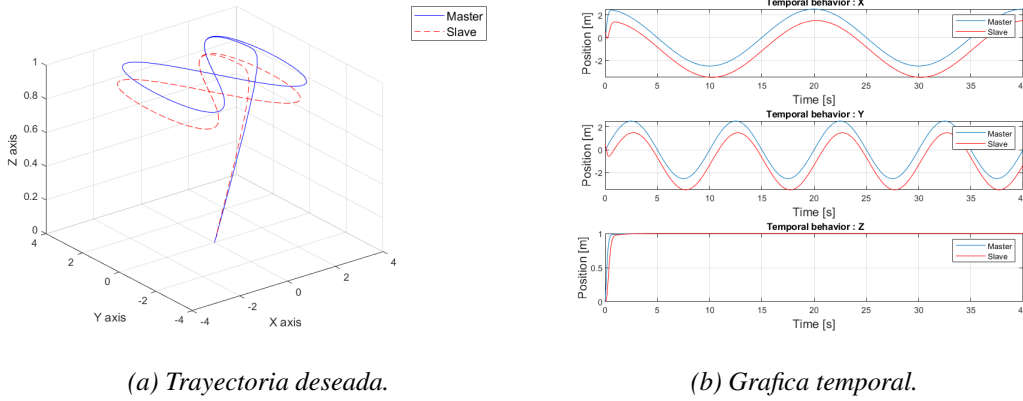


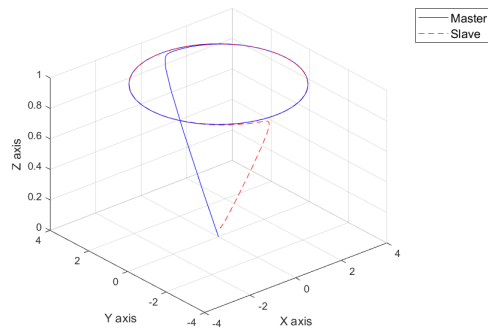
Figura 5.4: Control cuadricóptero individual: lemniscata.

Las Figuras 5.4 describen la trayectoria de una configuración maestro-esclavo, donde la línea punteada roja representa el cuadricóptero esclavo. Tenga en cuenta que $\Delta_x = \Delta_y = \Delta_z = 0$, y el ligero movimiento se debe al correcto funcionamiento de la etapa de anticollisiones.

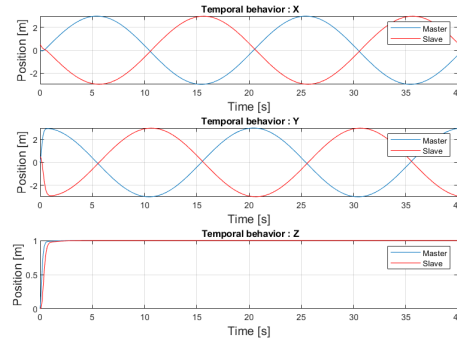
5.1.4 Configuración maestro-esclavo: control de trayectoria círculo

Siguiendo la Configuración maestro-esclavo ahora se aplica en trayectoria circular, en este ejemplo se aplica antifase en los ejes X, Y, con $\beta_5 = -1, \beta_6 = -1$.

La etapa de antifase se muestra en la figura 5.5, donde la línea punteada roja representa el cuadricóptero Esclavo que se encuentra en antifase con respecto al maestro.



(a) Trayectoria deseada.

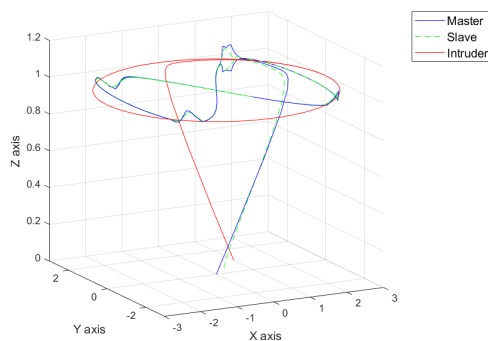


(b) Grafica temporal.

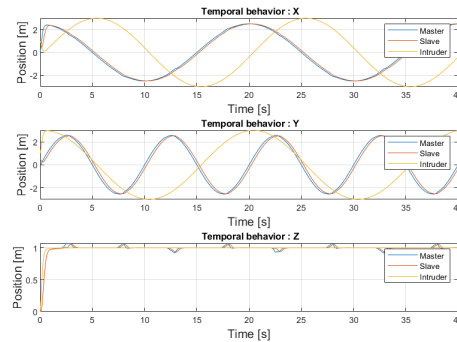
Figura 5.5: Configuración maestro-esclavo: círculo en antifase.

5.1.5 Configuración maestro-esclavo: Intruso

Para comprobar que el sistema reacciona a las anticollisiones, se introdujo un cuadricóptero Intruso, volando a la misma altura dibujando un círculo, como se muestra a continuación en la figura 5.6.



(a) Trayectoria deseada.



(b) Grafica temporal.

Figura 5.6: Configuración maestro-esclavo: intruso.

5.2 Trayectorias Concéntricas

5.2.1 Configuración maestro-esclavo concéntrica: control de trayectoria lemniscata

La siguiente simulación representa el comportamiento de la configuración concéntrica maestro-esclavo siguiendo una trayectoria de lemniscata, y el desplazamiento entre maestro y esclavo $\Delta_x = 0.5 * x_9$ y $\Delta_y = 0.5 * x_{11}$, estos valores se utilizan en las simulaciones siguientes.

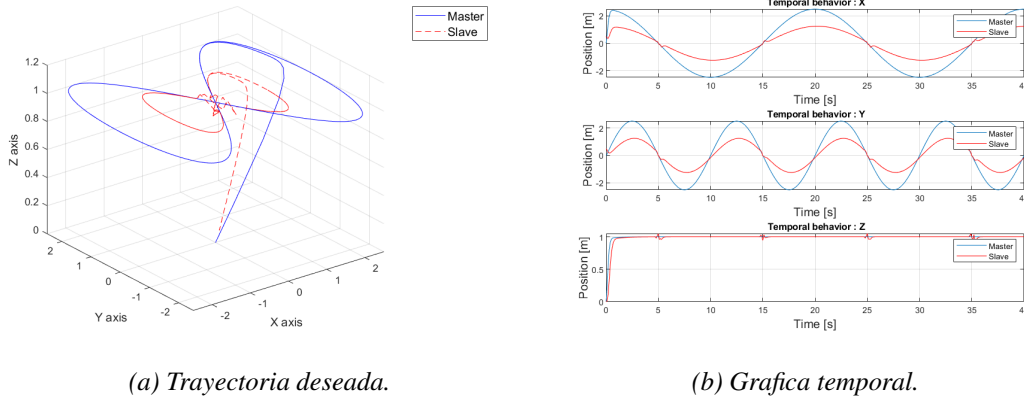
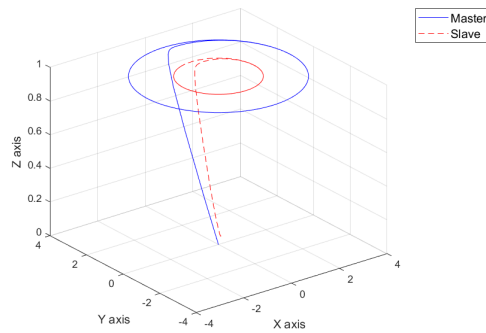


Figura 5.7: Configuración maestro-esclavo concéntrica: lemniscata.

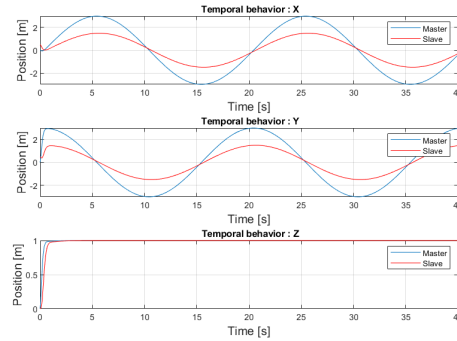
En la figura 5.7 se representa otro ejemplo, aquí mostramos una trayectoria concéntrica del cuadricóptero esclavo, por otro lado en la figura 5.7 b) se muestra como está funcionando la etapa anticoliisión en el eje Z.

5.2.2 Configuración maestro-esclavo concéntrica: control de trayectoria circulo

Ahora representando el comportamiento de la configuración concéntrica maestro-esclavo con una trayectoria circular, tenemos en la figura 5.8 que muestran una trayectoria circular concéntrica desde el cuadricóptero esclavo, en este ejemplo particular la etapa anticoliisión no se ve comprometida debido a la distancia entre los cuadricópteros.



(a) Trayectoria deseada.



(b) Grafica temporal.

Figura 5.8: Configuración maestro-esclavo concéntrica: círculo.

5.2.3 Configuración maestro-esclavo concéntrica: maestro 4 esclavos circular

En la figura 5.9, se muestra un ejemplo en el cual existe un maestro, seguido por 4 cuadricopteros esclavos de forma concéntrica, donde el cuadricóptero maestro está rodeado por los esclavos.

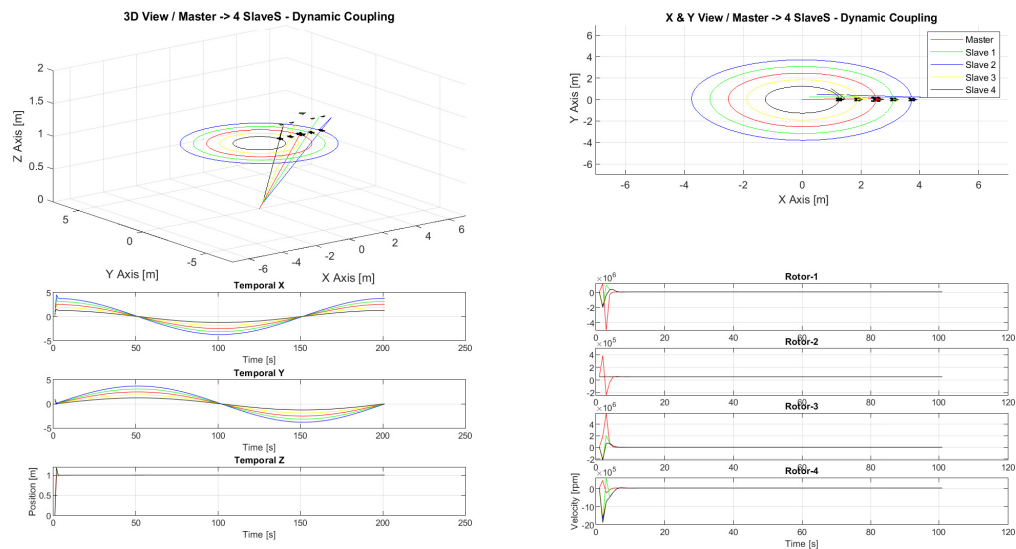


Figura 5.9: Simulación configuración maestro con 4 esclavos círculo.

5.2.4 Configuración maestro-esclavo concéntrica: maestro 4 esclavos lemniscata

En la figura 5.10, se muestra otra iteración del ejemplo anterior de un maestro seguido por 4 Esclavos, en este ejemplo en la trayectoria deseada a sido aplicada la de lemniscata, esto facilita ver el perfecto funcionamiento del anticoliciones en cada uno de los cuadricópteros.

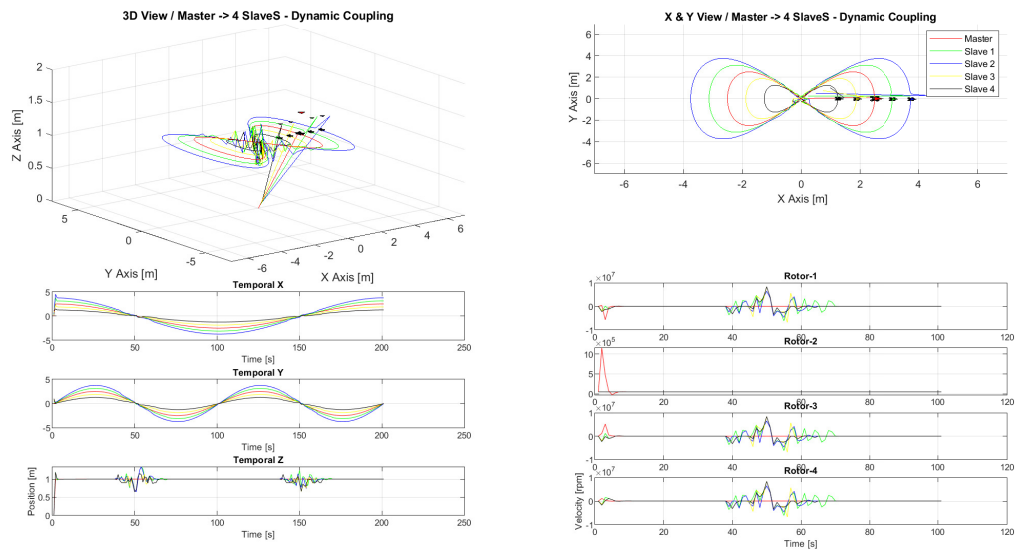


Figura 5.10: Simulación configuración maestro con 4 esclavos lemniscata.

5.3 Trayectorias caóticas

En algunas aplicaciones de la vida cotidiana, sobre todo en búsquedas inspeccionadas por robots, una de las mayores limitaciones es la cobertura que puede abarcar un robot siguiendo una ruta por determinado tiempo, es por esa razón; que es importante tener rutas alternativas que ayuden a reducir el tiempo de búsqueda de un objeto. Por lo anterior se propone una ruta caótica trazada por el maestro, es decir una formación que cubra un espacio delimitado por posiciones aleatorias (caóticas). Cabe mencionar que estas ideas son alternativas a trabajos realizados dentro del grupo de investigación, es decir se abordan de manera diferente. Para lograr lo anterior, se adecuo el sistema caótico de Lorenz discretizado mediante la difuminacion y dispersión de los puntos generados (función floor), los cuales serán dados como trayectoria deseada para un determinado cuadricóptero. El sistema se describe de la siguiente manera:

$$f(x,t) = a(y - x) \quad (5.3)$$

$$f(y,t) = x(b - z) - y \quad (5.4)$$

$$f(z,t) = xy - cz \quad (5.5)$$

Donde a, b y $c > 0$, en el sistema propuesto los valores son $a = 10$, $b = 28$ y $c = 8/3$. En la figura (5.11) se muestra el algoritmo diseñado para representar la arquitectura y estructura.

Programa creado en Matlab:

```
f1(1)=1;
f2(1)=1;
f3(1)=1;
caos = 1e-15;

if abs(f1(nc)-x(9))< caos abs(f2(nc)-x(11))< caos abs(f3(nc)-x(7))< caos
nc=nc+1;
```

$f1(nc)=(rand(1,1)*6)-2.5;$

$f2(nc)=(rand(1,1)*6)-2.5;$

$f3(nc)=(rand(1,1)*.5)+.75;$

end

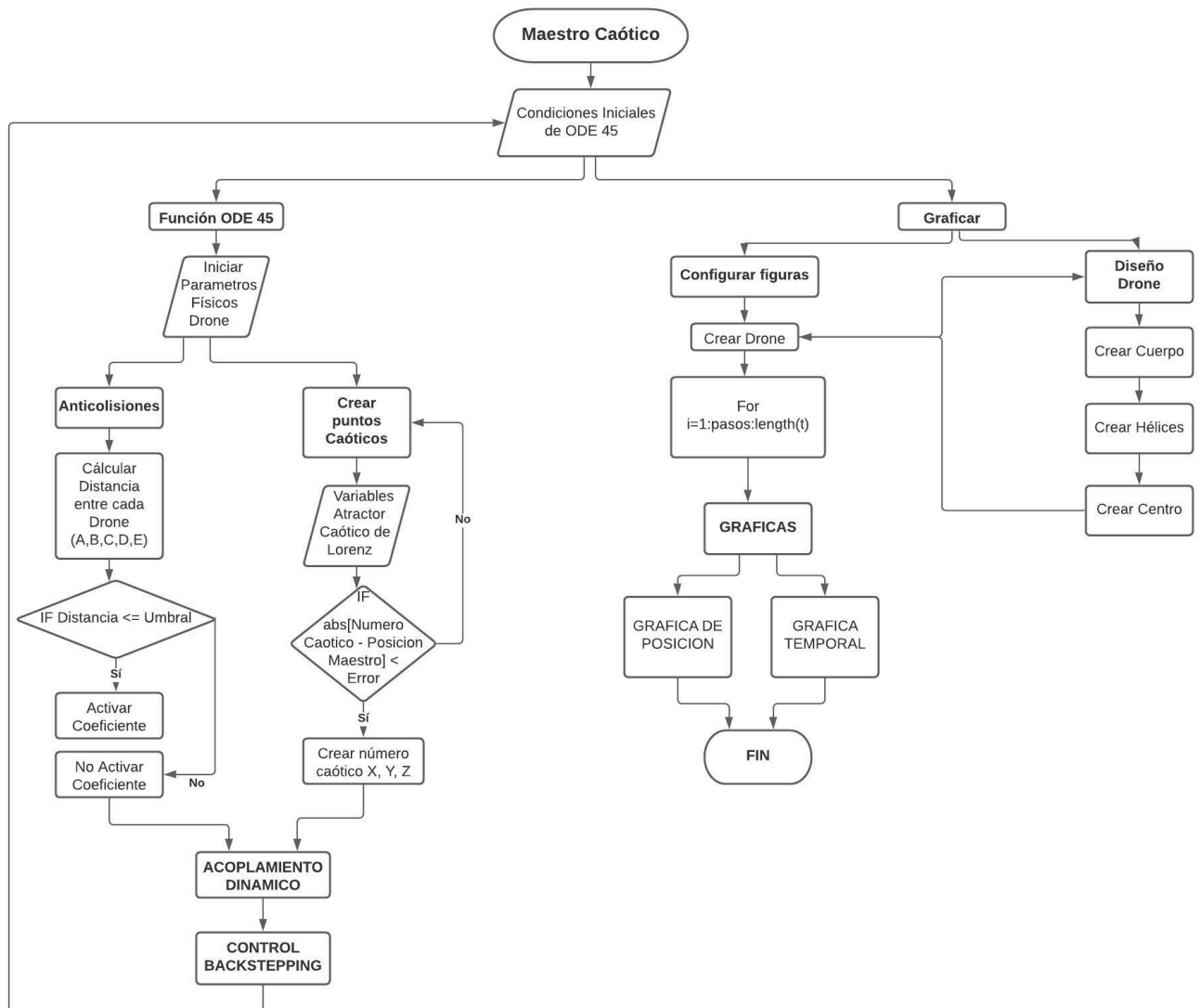


Figura 5.11: Diagrama de flujo - maestro caótico.

5.3.1 Configuración maestro-esclavo caótico

En los siguientes ejemplos se muestra cómo se comporta el sistema con las diferentes configuraciones que se tiene entre maestro – esclavo / intruso.

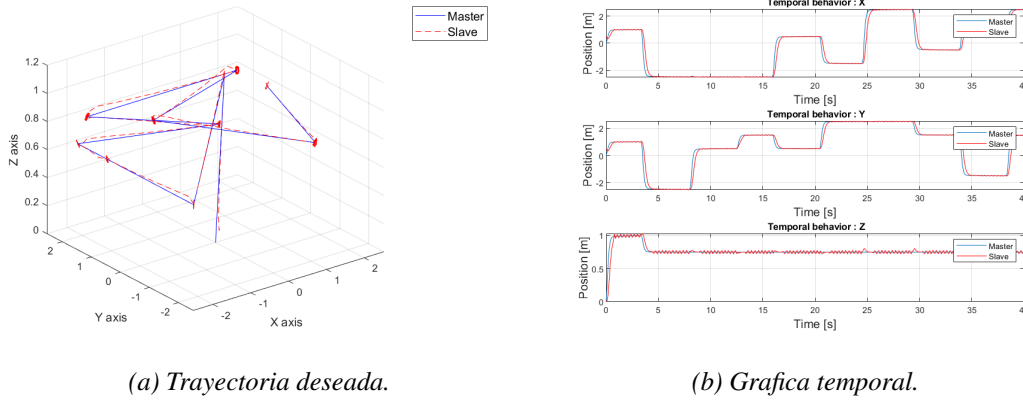


Figura 5.12: Configuración maestro-esclavo caótico.

5.3.2 Configuración maestro-esclavo caótico: antifase

En la figura 5.13 se puede observar como la línea azul representa el cuadricóptero maestro seguido por el Esclavo en antifase en los ejes X, Y y Z.

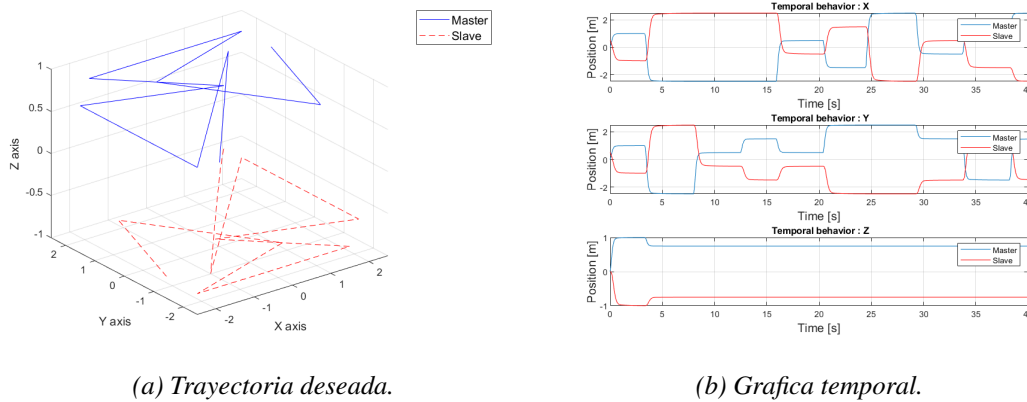
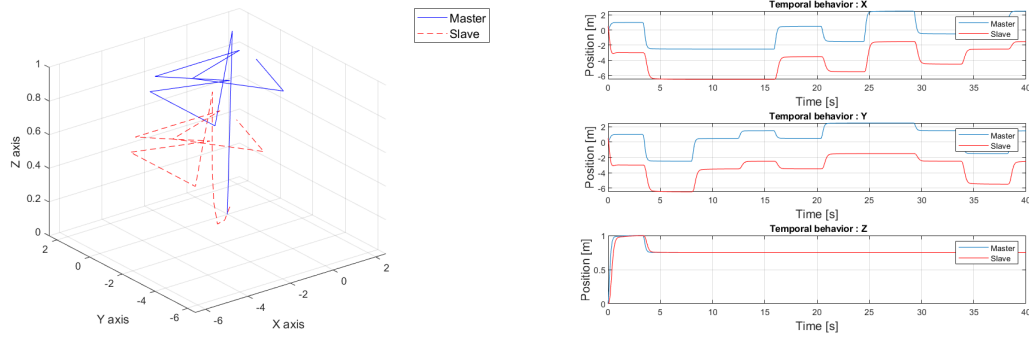


Figura 5.13: Configuración maestro-esclavo caótico: antifase.

5.3.3 Configuración maestro-esclavo caótico: formación

En la figura 5.14 se representa la configuración maestro-esclavo caótico en formación con valores $\Delta_x = 4$ y $\Delta_y = 4$.



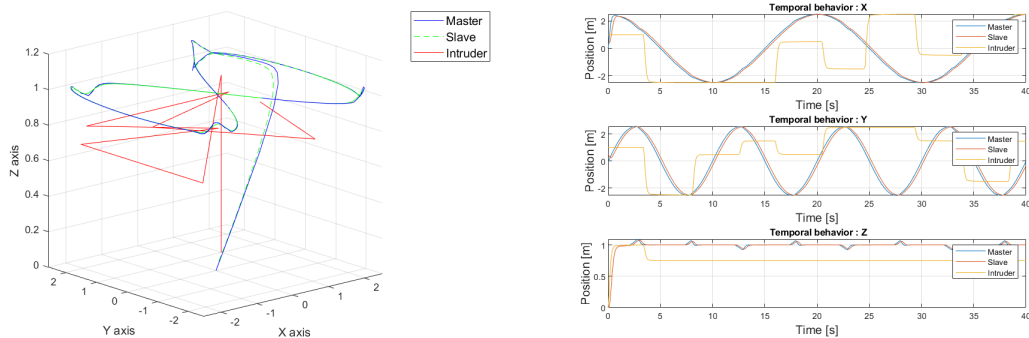
(a) Trayectoria deseada.

(b) Grafica temporal.

Figura 5.14: Configuración maestro-esclavo caótico: formado.

5.3.4 Configuración maestro-esclavo : intruso caótico

Este ejercicio representado en la figura 5.15, está diseñado con un comportamiento caótico, para demostrar que el sistema funciona con entradas aleatorias. Se utilizó una modificación de un atractor de Lorenz para crear posiciones aleatorias en el espacio.



(a) Trayectoria deseada.

(b) Grafica temporal.

Figura 5.15: Configuración maestro-esclavo: intruso caótico.

Capítulo 6

Maestro Intermitente Caótico

6.1 Algoritmo

En algunas aplicaciones de la vida diaria, especialmente en búsquedas donde se requiere inspección robótica, se necesita el diseño y ejecución de variaciones de algoritmos. Una de las mayores limitaciones es la cobertura que puede cubrir un robot siguiendo una ruta durante un tiempo determinado; esto significa que en ocasiones seguir un camino lineal no es eficiente para que el trabajo se realice correctamente; por ello, es importante contar con caminos alternativos que ayuden a reducir el tiempo de búsqueda. Por tanto, en este trabajo se propone un recorrido caótico trazado por un maestro, es decir, una formación que recorre un espacio delimitado por posiciones estocásticas (aleatorias). Para lograr lo anterior, se adaptó un sistema Lorenz caótico discretizado mediante desenfoque y dispersión de los puntos generados (usando la función floor en Matlab), el cual da una trayectoria deseada para determinado cuadricóptero. Un sistema caótico de Lorenz, es un sistema dinámico determinista tridimensional no lineal.

El "Algoritmo maestro intermitente caótico" es un diseño de algoritmo en este trabajo. El cual consiste en un maestro siguiendo una ruta caótica y un esclavo acompañándolo, en cada iteración ambos cuadricópteros tomarán la distancia entre el punto de su posición y el nuevo punto caótico, el cuadricóptero más cercano al siguiente punto caótico se convierte en el nuevo maestro del sistema.

$$DistMNA = \sqrt{(PcX - XA)^2 + (PcY - YA)^2 + (PcZ - ZA)^2} \quad (6.1)$$

Donde PcX, PcY y PcZ son los puntos de posición del punto caótico en el espacio, y XA, YA y ZA son los puntos de posición de un cuadricóptero. El algoritmo utilizado en este trabajo se describe para el diagrama que se muestra a continuación:

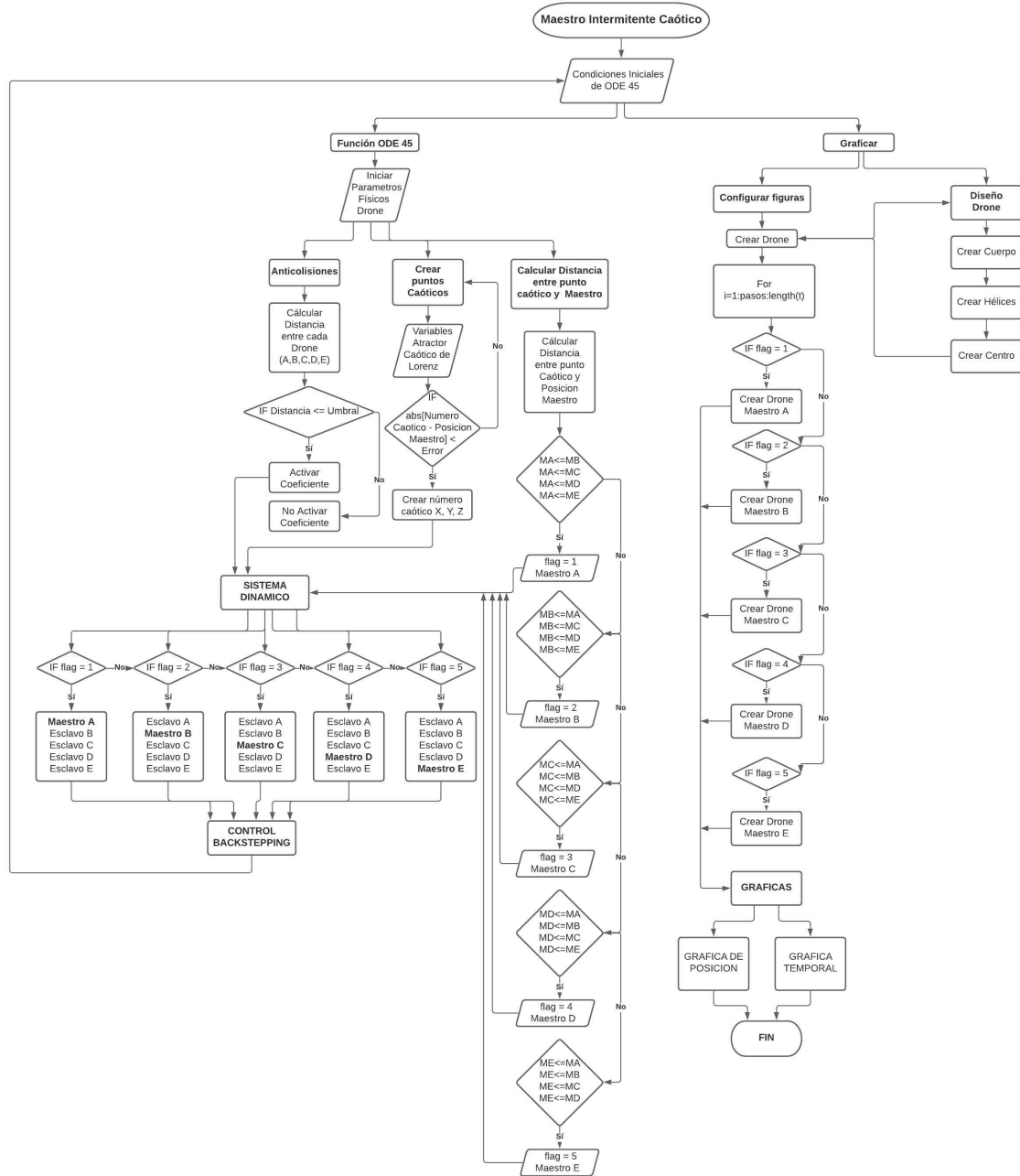


Figura 6.1: Diagrama de flujo maestro intermitente.

Algorithm: Intermittent Chaotic Master

Input = (x_0, t) : Initial Conditions of the quadcopter $(X_0, Y_0, Z_0, \phi_0, \theta_0, \psi_0)$, steps (time)

Output = $(X, Y, Z, \phi, \theta, \psi)$: Quadcopter position and rotation angles

1: *Run ODE 45*

2: *if* $\sqrt{(X_n - X_{n-i})^2 + (Y_n - Y_{n-i})^2 + (Z_n - Z_{n-i})^2} < Um$

$S = Um$;

$X_i = X_i * S$;

$Y_i = Y_i * S$;

$Z_i = Z_i * S$;

else:

$S = 0$;

return X_i, Y_i, Z_i ;

end

3: *If* $abs[(X_d - CP_X(i) < chaos \ \& \ Y_d - CP_Y(i) < chaos \ \& \ Z_d - CP_Z(i) < chaos]$

$Xh = a(y - x)$;

$Yh = x(b - z) - y$;

$Zh = xy - cz$;

$i++$;

$CP_X(i) = floor(((Xh(i) * 1000) - floor(Xh(i) * 1000)) * 6) - 2.5$;

$CP_Y(i) = floor(((Yh(i) * 1000) - floor(Yh(i) * 1000)) * 6) - 2.5$;

$CP_Z(i) = floor(((Zh(i) * 1000) - floor(Zh(i) * 1000)) * 0.5) + 0.75$;

$X_d = CP_X(i)$;

$Y_d = CP_Y(i)$;

$Z_d = CP_Z(i)$;

return X_d, Y_d, Z_d ;

end

4: *if* $DM(n) < DM(n - i)$

$flag = n$;

else:

$flag = n - i$;

return $flag$;

end

5: *Run Controller Equations*

6: *if* $flag = n$;

Run Dynamic Coupling Equations

return $X, Y, Z, \phi, \theta, \psi$;

end

7: *for* $k = 1: length(t)$:

Plot Graphs($X, Y, Z, \phi, \theta, \psi$);

Make Animations($X, Y, Z, \phi, \theta, \psi$);

end

El procedimiento que sigue el algoritmo maestro caótico intermitente es el siguiente: *paso 1*, Se inicializan todas las condiciones iniciales del sistema, es este caso se utiliza la función ODE 45 de Matlab para resolver las Ecuaciones Diferenciales; *paso 2*, se calculan las distancias entre los cuadricópteros para activar la etapa de anticollisiones en cada iteración que sea requerida; *paso 3*, se crean los puntos caóticos que después serán utilizados como las posiciones deseadas del cuadricóptero maestro; *paso 4*, se calcula la distancia entre cada uno de los cuadricópteros con respecto al punto caótico creado, en este punto se toma la decisión de cual cuadricóptero se convertirá en maestro; *paso 5*, se resuelven las ecuaciones diferenciales del controlador backstepping, cálculo de errores y entradas de control; *paso 6*, el controlador toma el valor de la bandera creada en el paso 4 para resolver las ecuaciones diferenciales del acoplamiento dinámico; *paso 7*, en un ciclo se ordenan y sincronizan los datos con el tiempo de reloj y se grafican los resultados finales.

A continuación, se presenta el comportamiento del algoritmo maestro intermitente caótico, por medio de simulaciones de las figuras 6.2 a 6.7, en el ejemplo se simulan 5 cuadricópteros, cada uno con marcadores en color rojo, verde, azul, amarillo y negro; para distinguirlos entre sí y un punto rojo que representa el punto caótico creado en el espacio. En la figura 6.2 se puede observar como los cuadricópteros salen de su posición inicial donde el cuadricóptero con marcador rojo es el maestro.

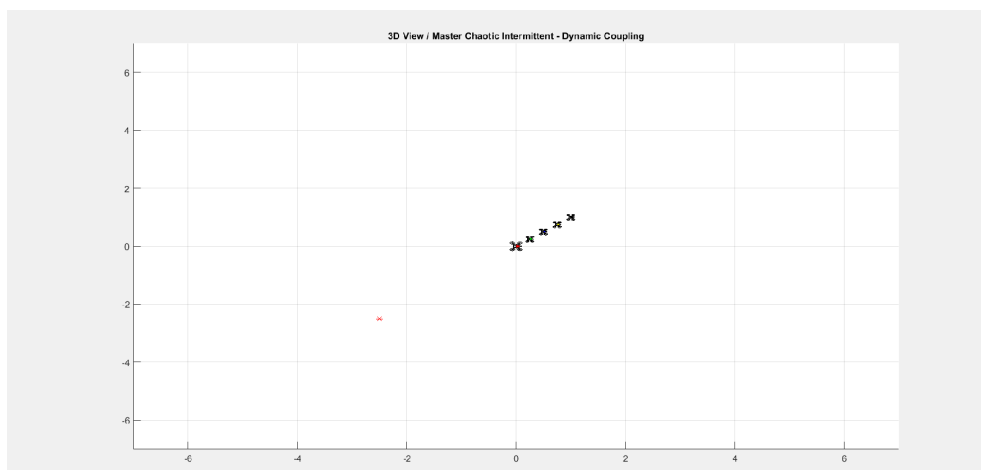


Figura 6.2: Simulación maestro-intermitente 1.

Logrando la formación en estrella como se observa en la figura 6.3, una vez que el maestro alcanza el punto caótico dibujado, se crea otro y el cuadricóptero más cercano en este caso

el cual tiene marcador azul se convierte automáticamente en maestro como se muestra en la figura 6.4.

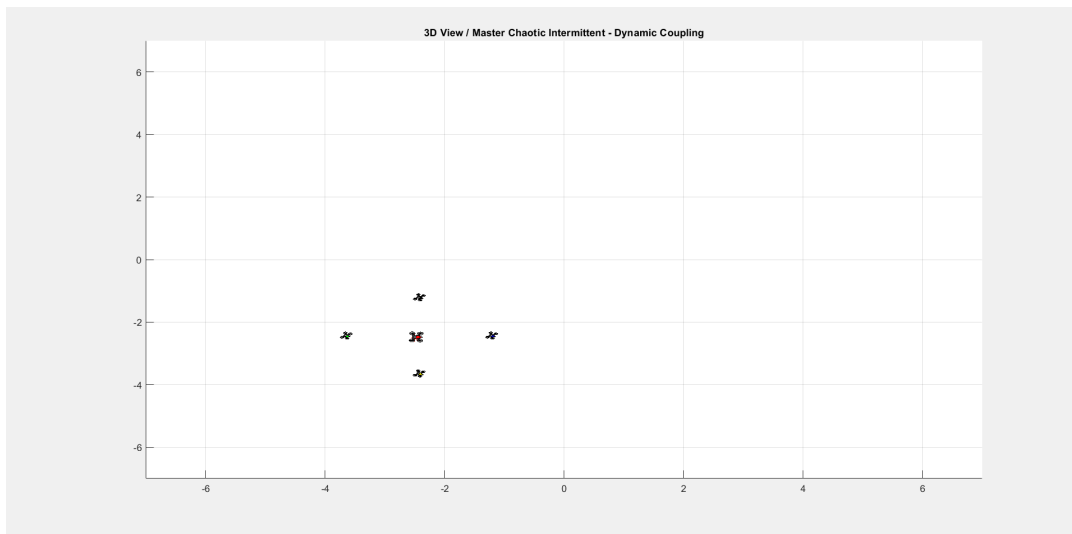


Figura 6.3: Simulación maestro-intermitente 2.

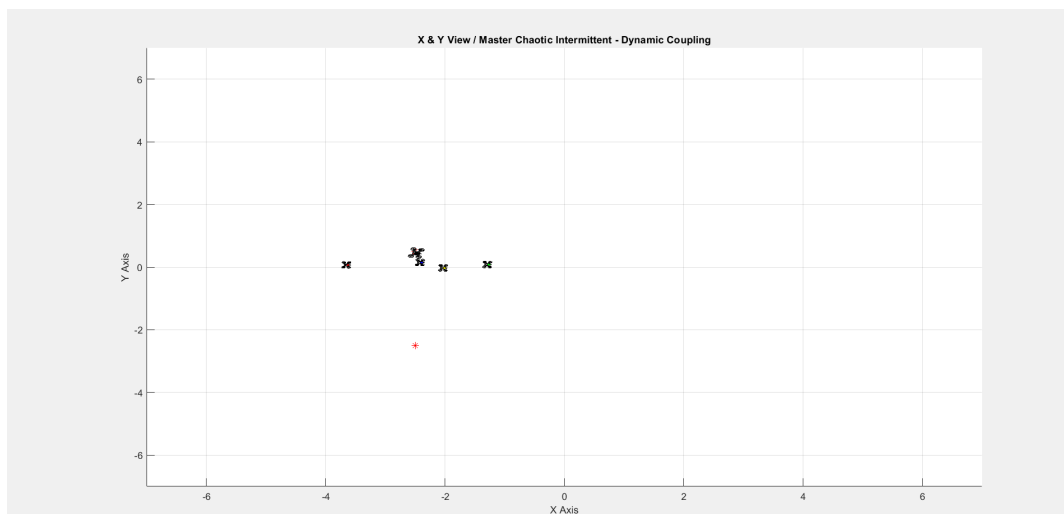


Figura 6.4: Simulación maestro-intermitente 3.

En la figura 6.5 ahora el cuadricóptero maestro es el que contiene el marcador verde.

En la figura 6.6 se observa que el maestro cambio a ser el marcado en color amarillo; por ultimo en la figura 6.7 se muestra como se crea otro punto caótico que convertirá al cuadricóptero más cercano en maestro, en este caso el verde regresa a ser maestro.

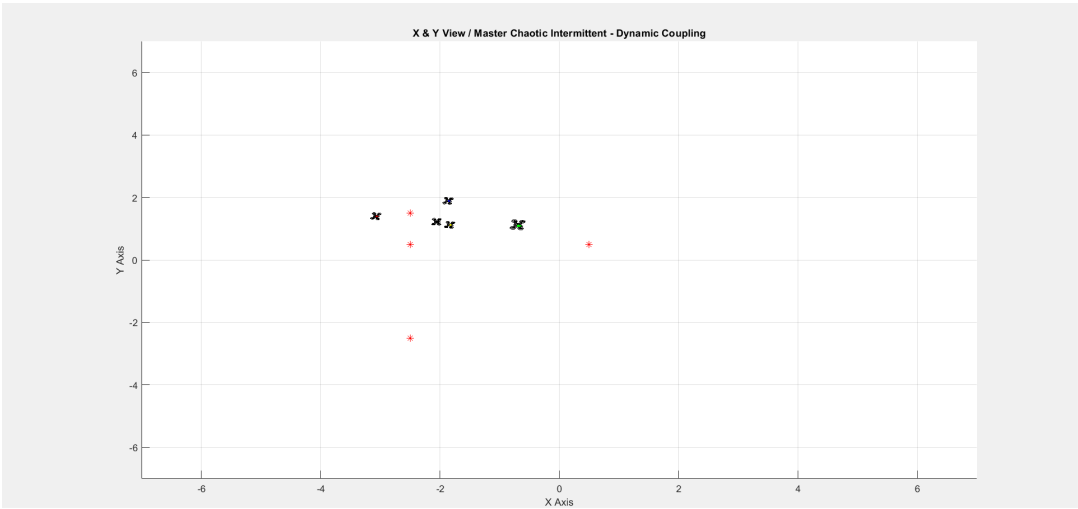


Figura 6.5: Simulación maestro-intermitente 4.

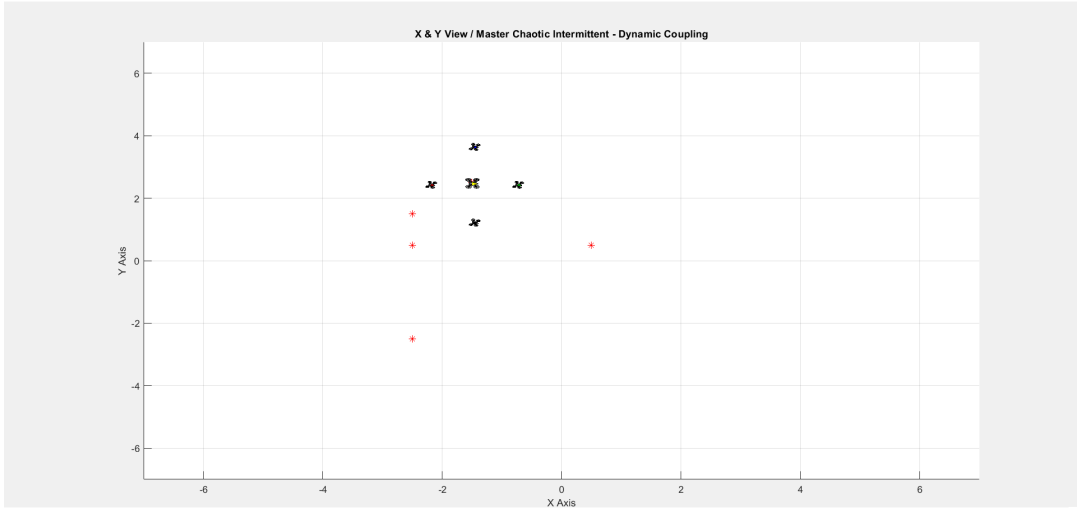


Figura 6.6: Simulación maestro-intermitente 5.

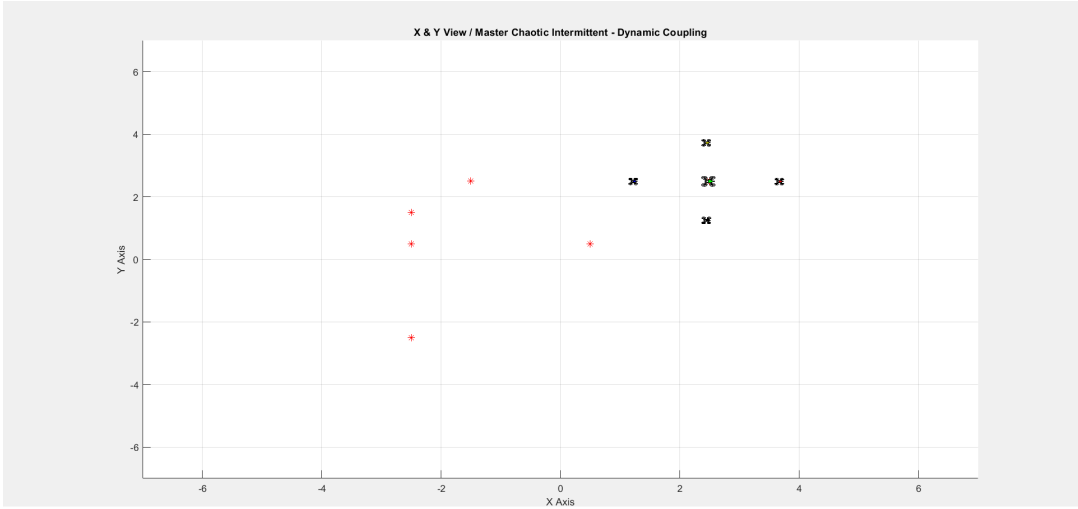


Figura 6.7: Simulación maestro-intermitente 6.

6.2 Simulaciones

Para representar los ejercicios de manera didáctica y visual, se tuvieron en cuenta varios métodos, entre los cuales estaban utilizar Matlab en conjunto con un API en este caso V-rep, mandando los datos de Matlab a V-rep, pero al estar diseñando un controlador para cuadricóptero las simulaciones se complicaron y no se obtuvo los resultados deseados; por lo que se optó por diseñar un programa en Matlab el cual crea animaciones las cuales representan de mejor forma lo que se desea plasmar. A continuación, se muestra un fragmento del código utilizado en Matlab para las simulaciones.

```

pasos = 5;
fig = figure;
set(fig,'position',[10 60 980 600]);
axis square; cameratoolbar
grid on;
axis([-3 3 -3 3 0 2]);
M1 = Diseno(y9(1),y11(1),y7(1),y1(1),y3(1),y5(1),6);
M2 = Diseno(y21(i),y23(i),y19(i),y13(i),y15(i),y17(i),6);
for i=1:pasos:length(t)
delete (M1)
delete (M2)
M1 = Diseno(y9(i),y11(i),y7(i),y1(i),y3(i),y5(i),6);
hold on, plot3(y9(1:i),y11(1:i),y7(1:i), '-b');
M2 = Diseno(y21(i),y23(i),y19(i),y13(i),y15(i),y17(i),6);
hold on, plot3(y21(1:i),y23(1:i),y19(1:i), '-g');
pause(0.1)
end
title(' Posicion / Intruso - Acoplamiento Dinamico ')
xlabel(' EJE X ')

```

```
ylabel(' EJE Y ')
```

```
zlabel(' EJE Z ')
```

```
legend('Maestro', 'Esclavo')
```

Capítulo 7

Conclusiones

En este trabajo, se concluye que usar controlador backstepping con acoplamiento dinámico es una mejor opción solo usar solo un controlador backstepping, aunque las simulaciones entre ambos sistemas son similares, existen las siguientes ventajas que refuerzan la idea de este trabajo.

1. Sistema intermediario, cada cómputo y cálculo pasa primero por el acoplamiento dinámico.
2. Las magnitudes de las entradas son menores después de pasar por el acoplamiento dinámico.
3. Los picos que son caudados por las entradas son filtrados por el sistema dinámico, que protege los equipos físicos en la implementación experimental.

En cuanto a las aplicaciones en las que se implementó el controlador propuesto, se demostró que el controlador cumple con la función que crear trayectorias suaves, convencionales, concéntricas y caóticas. El desarrollo de algoritmos para resolver situaciones en la vida cotidiana y científica; fue la principal idea por la que se desarrolló el algoritmo de maestro intermitente, en el futuro este algoritmo puede ser aplicado para realizar aplicaciones más específicas en el campo tecnológico, científico y militar.

7.1 Aportaciones

A continuación, se enlistan algunas de las aportaciones que este proyecto pone a disposición de la comunidad:

1. Implementación de un controlador backstepping y un acoplamiento dinámico para la formación de cuadricópteros.
2. Etapa de anticolisiones implementada en un grupo de cuadricópteros.
3. Creación de rutas caóticas en cuadricópteros.
4. Algoritmo maestro intermitente caótico.
5. Creación de scripts para simulaciones y animaciones.

7.2 Trabajo Futuro

Aunque en este trabajo se alcanzaron los objetivos marcados, existen mejoras o aportaciones que pueden ser añadidas en el futuro:

1. Implementación experimental.
2. Realizar sincronización y formación externa de robots móviles aéreos utilizando un acoplamiento dinámico enviado de manera segura.
3. Automatizar los scripts para agregar n-cantidad de robots.
4. Implementar otros tipos de atractores caóticos en la generación de rutas caóticas.
5. Utilizar este trabajo para crear aplicaciones en la vida real, combinando técnicas y algoritmos de inteligencia artificial.

Bibliografía

- [1] A. McKenna. (2013). The future of Drone use: Opportunities and Threats from Ethical and Legal Perspectives,. United Kingdom: Asser Press - Springer, p. 355.
- [2] O.E.U. (2015). Community Research and Development Information Service, Retrieved October 9 2020.
- [3] Floreano D. (2015). Science, technology and the future of small autonomous drones. Nature. 521 (7553): 460–466. Bibcode:2015Natur.521..460F. doi:10.1038/nature14542. PMID 26017445. S2CID 4463263.
- [4] Carlson, Daniel F., (2018). Adapting open-source drone autopilots for real-time iceberg observations. MethodsX, p. 1059-1072.
- [5] Ackerman, Spencer, Shachtman, Noah. (2012). Almost 1 in 3 U.S. Warplanes Is a Robot. Retrieved from WIRED, 9 October 2020.
- [6] Peck A., (2020). Coronavirus Spurs Perceptor's Drone-in-a-box Surveillance Solution. DRONELIFE, retrieved 9 October 2020.
- [7] H. Bouadi, M. Bouchoucha, and M. Tadjine. Sliding Mode Control based on Backstepping Approach for an UAV Type-Quadrotor. World Academy of Science, Engineering and Technology International Journal of Mechanical, Aerospace, Industrial, Mechatronic and Manufacturing Engineering Vol:1, No:2, 2007. 2.1
- [8] Ashfaq Ahmad Mian, Wang Daobo (2008). Modeling and Backstepping-based Nonlinear Control Strategy for a 6 DOF Quadrotor Helicopter. Chinese Journal of Aeronautics.

- Volume 21, Issue 3, pp. 261-268, ISSN 1000-9361. Recuperado el 13 de octubre del 2020, de: <https://www.sciencedirect.com/science/article/pii/S1000936108600345>. 2.1
- [9] Sahul M.P.V, V. Naveen C., and Thomas K. (2014). A novel method on Disturbance Rejection PID Controller for Quadcopter based on Optimization algorithm. Third International Conference on Advances in Control and Optimization of Dynamical Systems. pp. 192-199, 10.3182/20140313-3-IN-3024.00016. Recuperado el 13 de octubre del 2020, de: <https://pdf.sciencedirectassets.com/314898>. 2.1
- [10] EDO USM (2019). Aplicación del metodo Lyapunov en un ejemplo. Recuperado el 27 de Octubre del 2020, de: <https://www.youtube.com/watch?v=D-tnLvqr10> 2.1
- [11] Abdul-Wahid A. Saif, Mujahid Dhaifullah, Mohammad Al-Malki Mustafa El Shafie (2012). Modified Backstepping Control of Quadrotor. IEEE. No. 12726672. Recuperado el 13 de octubre del 2020, de: <https://ieeexplore.ieee.org/abstract/document/6197975> 2.1
- [12] Akhilesh S. (2014). Comparison of Quadrotor Performance Using Backstepping and Sliding Mode Control. International Conference on Circuits, Systems and Control. pp. 79-82, ISBN: 978-1-61804-216-3. Recuperado el 13 de octubre del 2020, de: <https://www.semanticscholar.org/paper/Comparison-of-Quadrotor-Performance-Using-and-Mode-Swarup/5ce0499ea070bc6a33938bd2a2884cad53c6c43e> 2.1
- [13] S. Bouabdallah and R. Siegwart. (2005). Backstepping and Sliding-mode Techniques Applied to an Indoor Micro Quadrotor. pp. 2247-2252, doi: 10.1109/ROBOT.2005.1570447. Recuperado el 13 de octubre del 2020, de: <https://ieeexplore.ieee.org/document/1570447> 2.1
- [14] Emilio M. (2016). Nonlinear Model Predictive Control of a Quadrotor. pp. 2247-2252, doi: 10.1109/ROBOT.2005.1570447. Tesis para grado de Maestría, Escola Tècnica Superior d'Enginyeria Industrial de Barcelona. Recuperado el 13 de octubre del 2020, de: <https://upcommons.upc.edu/bitstream/handle/2117/98503> 2.1
- [15] Arshad M., Yoonsoo K. (2017). Decentralized formation flight control of quadcopters using robust feedback linearization. Journal of the Franklin Institute.

- pp. 852-871, ISSN 0016-0032. Recuperado el 13 de octubre del 2020, de: <https://www.sciencedirect.com/science/article/abs/pii/S0016003216304070> 2.1
- [16] Nguyen Xuan-Mung, Sung Kyung-Hong. (2019). Robust adaptive formation control of quadcopters based on leader-follower approach. *International Journal of Advanced Robotic Systems*. Volume 16, Issue 4, doi: 1729881419862733. Recuperado el 13 de octubre del 2020, de: <https://journals.sagepub.com/doi/full/10.1177/1729881419862733> 2.1
- [17] Alexandre H., Wagner T., Maria G., Edson P., Murilo Z. Tamires dos Santos. (2020). Decentralized Multi-Robot System for Formation Control of Quadcopters: The Integration Between the Virtual and Real World. 17th International Conference on Information Technology-New Generations. Print ISBN: 978-3-030-43019-1. Recuperado el 13 de octubre del 2020, de: <https://link.springer.com/chapter/10.1007/978-3-030-43020-7> 2.1
- [18] Muslimov TZ, Munasypov RA. (2020). Adaptive decentralized flocking control of multi-UAV circular formations based on vector fields and backstepping. *ISA Trans.* 2020 Dec;107:143-159. doi: 10.1016/j.isatra.2020.08.011. Epub 2020 Aug 21. PMID: 32863052. Recuperado el 13 de octubre del 2020, de: <https://pubmed.ncbi.nlm.nih.gov/32863052/> 2.1
- [19] EDO USM (2019). Aplicación del metodo Lyapunov en un ejemplo. Recuperado el 27 de Octubre del 2020, de: <https://www.youtube.com/watch?v=D-tnLvrq10> 2.2.2
- [20] Kotkovic P. (1992). The joy of feedback: nonlinear and adaptive. *IEEE control magazibe*. 12(3): 7-17. 2.2.2
- [21] I. Fantoni, R. Lozano, *Nonlinear Control for Underactuated Mechanical Systems* (Springer, New York, 2001) 2.2.4, 1
- [22] Posadas C. (2008). Sincronización de Osciladores de Lorenz por formas Hamiltonianas. Tesis Maestría en Ciencias CICESE. 3
- [23] Steven Strogatz. *Sync*. Ed New York

- [24] S.N. (2022). CrazyFly, Silicio.mx, retrieved from: <https://silicio.mx/crazyflie-20> , January 8, 2022.
- [25] York Vara Herrera. (Master Degree Tesis 2021). Diseño de Trayectorias para Robots Móviles Tipo Diferencial. Universidad Autónoma de Baja California.
- [26] Pena Ramirez, J., Arellano-Delgado, A., y Nijmeijer, H. (2018). Enhancing master-slave synchronization: The effect of using a dynamic coupling. *Physical Review E*, 98(1): 1–10.
- [27] López Parra A. (Master Degree Tesis 2017). Formación en grupos de robots móviles. Centro de Investigación Científica y de Educación Superior de Ensenada.
- [28] López-Parra A., Arellano-Delgado A., López-Gutiérrez R.M., Cruz-Hernández C., Formation in networks in mobile robots. *Proceedings of the Eighth International Conference on Complex Systems Design and Management, CSD-M Paris 2017*.
- [29] A. López-Parra, A. Arellano-Delgado, C. Cruz-Hernández, R. Martínez-Clark, J.J. Cetina-Denis. Formación en grupos de robots móviles. *Latin American Conference on Automatic Control*, 13 a 15 de octubre de 2016, Medellín, Colombia.
- [30] Erik Hernández Valente. (Master Degree Tesis 2018). Sincronización de robots terrestres y aéreos. Universidad Autónoma de Baja California.
- [31] Wang X.F. Complex networks: Toplogy, dynamics and synchronization. *International Journal Bifurcation and Chaos*, 12(5), (2002) 885-916.
- [32] Khalil H. (1996) *Nonlinear systems*. Prentice Hall.
- [33] Fradkov A., Miroshnik I.V. y Nikiforov O. (1999) *Nonlinear and adaptive control of complex systems*. Kluwer Academic Publishers.
- [34] Martinez C. (2014). Diseño de un controlador para la formación de grupos de robots móviles. Universidad Autónoma de Baja California. 3.4

Anexos

A continuación, se muestra la sección de anexos: Anexo 1: Código para la simulación del control y acoplamiento de la configuración Maestro-Esclavo-Intruso.

```

function SimulacionMaestroEsclavo()
% Utilizar control por backstepping para hacer formacion de un drone
% Sistemas Dinamicos Maestro & Esclavo con Intruso
% Se agrego un anticolisiones en el esclavo
% Seguir trayectorias Caoticas Maestro - Esclavo(acoplamiento dinamico)
% Seguir trayectorias Convencionales
% Seguir trayectorias Concentricas
% Se agrego las velocidades de los rotores
% Autor: Johnny Rubio Pecasso
% Version: 4.3
% Octubre/27/2021
% Actualizacion: Noviembre/26/2021
clear;
clc;
close all;

P = 20;
global it;
it = 1;
global contador;
contador = 1;
global tiempo;
tiempo = 1;
pas = 0.1;
global Gx
global Gy
global Gz
global x1d;
global x3d;
global x1dI;
global x3dI;
x1d=0;
x3d=0;
global WWA;
global WV1A;
global WV2A;
global WV3A;
global WV4A;
global WWB;
global WV1B;
global WV2B;
global WV3B;
global WV4B;
global WWI;
global WV1I;
global WV2I;
global WV3I;
global WV4I;

% Condiciones iniciales

```

[illegible]

```

        WVv4A(j) = WV4A(i);

        WVv1B(j) = WV1B(i);
        WVv2B(j) = WV2B(1);
        WVv3B(j) = WV3B(i);
        WVv4B(j) = WV4B(i);

        WVv1I(j) = WV1I(i);
        WVv2I(j) = WV2I(1);
        WVv3I(j) = WV3I(i);
        WVv4I(j) = WV4I(i);

        conteo1 = conteo1 + pas;
        conteo2 = conteo2 + pas;
        tm(j) = j;
        j = j + 1;
    end
end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Simulacion de drones
pasos = 1;
fig = figure;
set(fig,'position',[0 0 2160 1080]);

subplot(6,2,[1,3]);
M1 = Diseno(AX(1),AY(1),AZ(1),A1(1),A2(1),A3(1),6,'b');
M2 = Diseno(BX(1),BY(1),BZ(1),B1(1),B2(1),B3(1),6,'g');
M3 = Diseno(IX(1),IY(1),IZ(1),I1(1),I2(1),I3(1),6,'r');
grid on;
axis square; cameratoolbar
axis([-7 7 -7 7 0 2]);

subplot(6,2,[2,4]);
M4 = Diseno(AX(1),AY(1),0,A1(1),A2(1),A3(1),6,'b');
M5 = Diseno(BX(1),BY(1),0,B1(1),B2(1),B3(1),6,'g');
M6 = Diseno(IX(1),IY(1),0,I1(1),I2(1),I3(1),6,'r');
grid on;
axis([-7 7 -7 7]);

pause(1)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

for i=1:pasos:length(t)
    delete (M1)
    delete (M2)
    delete (M3)
    delete (M4)

```

```
delete (M5)
delete (M6)
```

```
subplot(6,2,[1,3]);
axis([-7 7 -7 7 0 2]);
M1 = Diseno(AX(i),AY(i),AZ(i),A1(i),A2(i),A3(i),9,'b');
hold on;
grid on;
plot3(AX(1:i),AY(1:i),AZ(1:i), '-b');
M2 = Diseno(BX(i),BY(i),BZ(i),B1(i),B2(i),B3(i),6,'g');
hold on;
grid on;
plot3(BX(1:i),BY(1:i),BZ(1:i), '-g');
M3 = Diseno(IX(i),IY(i),IZ(i),I1(i),I2(i),I3(i),9,'r');
hold on;
grid on;
plot3(IX(1:i),IY(1:i),IZ(1:i), '-r');
title(' 3D View / Master -> Slave / Intruder ')
xlabel(' X Axis [m]')
ylabel(' Y Axis [m]')
% legend('Master', 'Slave', 'Intruder')
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

```
subplot(6,2,[2,4]);
axis([-7 7 -7 7]);
M4 = Diseno(AX(i),AY(i),AZ(i),A1(i),A2(i),A3(i),9,'b');
hold on;
grid on;
plot(AX(1:i),AY(1:i), '-b');
M5 = Diseno(BX(i),BY(i),BZ(i),B1(i),B2(i),B3(i),6,'g');
hold on;
grid on;
plot(BX(1:i),BY(1:i), '-g');
M6 = Diseno(IX(i),IY(i),IZ(i),I1(i),I2(i),I3(i),6,'r');
hold on;
grid on;
plot(IX(1:i),IY(1:i), '-r');
title(' X & Y View / Master -> Slave / Intruder ')
xlabel(' X Axis [m]')
ylabel(' Y Axis [m]')
legend('Master', 'Slave', 'Intruder')
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

```
subplot(6,2,7);
plot(AX(1:i),'b');
hold on;
plot(BX(1:i), 'g');
hold on;
plot(IX(1:i), 'r');
```

```
hold on;
grid on
title(' Temporal X ')
xlabel('Time [S]')
```

```
subplot(6,2,9);
plot(AY(1:i),'b');
hold on;
plot(BY(1:i), 'g');
hold on;
plot(IY(1:i), 'r');
hold on;
grid on
title(' Temporal Y ')
xlabel('Time [S]')
```

```
subplot(6,2,11);
plot(AZ(1:i),'b');
hold on;
plot(BZ(1:i), 'g');
hold on;
plot(IZ(1:i), 'r');
hold on;
grid on
title(' Temporal Z ')
xlabel('Time [S]')
ylabel('Position [m]')
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

```
subplot(8,2,10);
plot(WVv1A(1:i),'b');
hold on;
plot(WVv1B(1:i),'g');
hold on;
plot(WVv1I(1:i),'r');
hold on;
grid on;
title(' Rotor-1 ')
```

```
subplot(8,2,12);
plot(WVv2A(1:i),'b');
hold on;
plot(WVv2B(1:i),'g');
hold on;
plot(WVv2I(1:i),'r');
hold on;
grid on;
title(' Rotor-2 ')
```

```
subplot(8,2,14);
```

```

    plot(WVv3A(1:i),'b');
    hold on;
    plot(WVv3B(1:i),'g');
    hold on;
    plot(WVv3I(1:i),'r');
    hold on;
    grid on;
    title(' Rotor-3 ')

    subplot(8,2,16);
    plot(WVv4A(1:i),'b');
    hold on;
    plot(WVv4B(1:i),'g');
    hold on;
    plot(WVv4I(1:i),'r');
    hold on;
    grid on;
    title(' Rotor-4 ')
    xlabel('Time [s]')
    ylabel('Velocity [rpm]')
    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    pause(1e-24)
end

% %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% % Graficas de fase
% set(figure,'position',[800 220 720 500])
% subplot(3,1,1)
% plot(ts,y(:,9),ts,y(:,21),ts,y(:,33));
% grid on
% title(' Temporal behavior : X')
% xlabel('Time [s]','FontSize', 12)
% ylabel('Position [m]','FontSize', 12)
% legend('Master', 'Slave', 'Intruder')
%
% subplot(3,1,2)
% plot(ts,y(:,11),ts,y(:,23),ts,y(:,35));
% grid on
% title(' Temporal behavior : Y')
% xlabel('Time [s]','FontSize', 12)
% ylabel('Position [m]','FontSize', 12)
% legend('Master', 'Slave', 'Intruder')
%
% subplot(3,1,3)
% plot(ts,y(:,7),ts,y(:,19),ts,y(:,31));
% grid on
% title(' Temporal behavior : Z')
% xlabel('Time [s]','FontSize', 12)
% ylabel('Position [m]','FontSize', 12)

```

```

% legend('Master', 'Slave', 'Intruder')
% % ylim([-0.5 0.5])
%
% %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% % Graficas de seguimiento y ruta
% set(figure,'position',[60 220 720 500])
% plot3(AX,AY,AZ,'-b') % Posicion Maestro
% hold on
% plot3(BX,BY,BZ,'--g') % Posicion Esclavo
% grid on
% plot3(IX,IY,IZ,'-r') % Posicion Intruso
% grid on
% % title(' Master-Slave Configuration ')
% xlabel(' X axis ','FontSize', 12)
% ylabel(' Y axis ','FontSize', 12)
% zlabel(' Z axis ','FontSize', 12)
% legend('Master', 'Slave', 'Intruder', 'FontSize', 12)
%
% %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% % Graficas de angulos
% set(figure,'position',[800 220 720 500])
% subplot(3,1,1)
% plot(ts,y(:,1),ts,y(:,13),ts,y(:,25),'g');
% grid on
% title(' Roll Angle')
% xlabel(' Time [s] ','FontSize', 12)
% ylabel(' Position [rad] ','FontSize', 12)
% legend('Master', 'Slave', 'Intruder')
%
% subplot(3,1,2)
% plot(ts,y(:,3),ts,y(:,15),ts,y(:,27),'g');
% grid on
% title(' Pitch Angle')
% xlabel(' Time [s] ','FontSize', 12)
% ylabel(' Position [rad] ','FontSize', 12)
% legend('Master', 'Slave', 'Intruder')
%
% subplot(3,1,3)
% plot(ts,y(:,5),ts,y(:,17),ts,y(:,29),'g');
% grid on
% title(' Yaw Angle')
% xlabel(' Time [s] ','FontSize', 12)
% ylabel(' Position [rad] ','FontSize', 12)
% legend('Master', 'Slave', 'Intruder')
% % ylim([-0.5 0.5])

mensaje = 'FIN DE LA SIMULACION';
disp(mensaje)
end

```

```

function dx=SistemaControl(t,x)
% Modelo de la funcion de control de cuadricoptero
% Parametros fisicos
b = 3.13e-5; % Thrust factor
d = 7.5e-7; % Drag factor
% Inercia
Ix = 7.5e-3;
Iy = 7.5e-3;
Iz = 1.3e-2;
Jr = 6e-5; % inercia motor
l = .23; % lever
b1 = l / Ix;
b2 = l / Iy;
b3 = l / Iz;
g = 9.81; % Gravedad
m = .650; % masa
P = 20; % Periodo

```

```

global it
global xxC
global yC
global zC
global f1
global f2
global f3
global nc
global contador
global Gx
global Gy
global Gz
global x1d;
global x3d;
global x1dI;
global x3dI;
global tiempo;
global WWA;
global WV1A;
global WV2A;
global WV3A;
global WV4A;
global WWWB;
global WV1B;
global WV2B;
global WV3B;
global WV4B;
global WWWI;
global WV1I;
global WV2I;
global WV3I;
global WV4I;

```

```

% Velocidades de los rotores
O1 = 15000; % Rotor 1
O2 = 15000; % Rotor 2
O3 = 15000; % Rotor 3
O4 = 15000; % Rotor 4
omega = O2 + O4 - O1 - O3;

% Variables
a1=(Iy-Iz)/Ix;
a2=(-Jr/Ix);
a3=(Iz-Ix)/Iy;
a4=(-Jr/Iy);
a5=(Ix-Iy)/Iz;

%-----
% Evacion de colisiones
% Evacion entre A-B
Rm = 0.25; % Coeficiente de anticollision
Um = 0.5; % Umbral de colision
DAB = sqrt( (x(19)-x(7))^2 + (x(21)-x(9))^2 + (x(23)-x(11))^2 ); % Distancia entre Drones
% Desicion para distancia entre drones
if DAB <= Um
    RABz = Rm;
    RABx = Rm;
    RABy = Rm;
else
    RABz = 0;
    RABx = 0;
    RABy = 0;
end

DAI = sqrt( (x(31)-x(7))^2 + (x(33)-x(9))^2 + (x(35)-x(11))^2 ); % Distancia entre Drones
% Desicion para distancia entre drones
if DAI <= Um
    RAIz = Rm;
    RAIx = Rm;
    RAIy = Rm;
else
    RAIz = 0;
    RAIx = 0;
    RAIy = 0;
end

DBI = sqrt( (x(19)-x(31))^2 + (x(21)-x(33))^2 + (x(23)-x(35))^2 ); % Distancia entre Drones
% Desicion para distancia entre drones
if DBI <= Um

```

```

        RBIz = Rm;
        RBIx = Rm;
        RBIy = Rm;
    else
        RBIz = 0;
        RBIx = 0;
        RBIy = 0;
    end
end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% %-----
% % Ruta caotica para Maestro
% xxC(1)=1;
% yC(1)=1;
% zC(1)=1;
% f1(1)=xxC(1);
% f2(1)=yC(1);
% f3(1)=zC(1);
% tao=0.01;
% aC=10;
% cC=8/3;
% bC=28;
% caos = 0.001; % Variable para aumentar las rutas deseadas caoticas
%
% if abs(f1(nc)-x(33))< caos && abs(f2(nc)-x(35))< caos && abs(f3(nc)-x(31))< caos
%     % Atractor Caotico de Lorenz
%     xxC(nc+1)=xxC(nc)+tao*(aC.*(yC(nc)-xxC(nc)));
%     yC(nc+1)=yC(nc)+tao*(xxC(nc)*(bC-zC(nc))-yC(nc));
%     zC(nc+1)=zC(nc)+tao*(yC(nc)*xxC(nc)-cC*zC(nc));
%
%     nc=nc+1;
%
%     f1(nc)=floor(((xxC(nc)*1000)-floor(xxC(nc)*1000))*6)-2.5;
%     f2(nc)=floor(((yC(nc)*1000)-floor(yC(nc)*1000))*6)-2.5;
%     f3(nc)=floor(((zC(nc)*1000)-floor(zC(nc)*1000))*0.5)+0.75;
%
% %     f1(nc)=(rand(1,1)*6)-2.5;
% %     f2(nc)=(rand(1,1)*6)-2.5;
% %     f3(nc)=(rand(1,1)*0.5)+0.75;
%
% end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%-----
% Sistema Maestro
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Parametros de Control deseadas
x5d = 0*(pi/180); % Yaw deseado
% x7d = f3(nc); % Posicion en Z
% x9d = f1(nc); % Posicion en X

```

```

% x11d = f2(nc); % Posicion en Y

% % Seguimiento de trayectoria en circulo
% x7d = 1; % Posicion en Z
% x9d = 3*sin(((2*pi)/P)*t); % Posicion en X
% x11d = 3*cos(((2*pi)/P)*t); % Posicion en Y

% % Seguimiento de trayectoria en espiral
% x7d = 0.1*t; % Posicion en Z
% x9d = 2.5*cos(((2*pi)/P)*t); % Posicion en X
% x11d = 2.5*sin(((2*pi)/P)*t); % Posicion en Y

% % Seguimiento de trayectoria en 8
x7d = 1; % Posicion en Z
x9d = 2.5*cos(((2*pi)/P)*t); % Posicion en X
x11d = 2.5*sin(((4*pi)/P)*t); % Posicion en Y

Gx(it) = x9d;
Gy(it) = x11d;
Gz(it) = x7d;

% Ganancias
ganA = 10;
alpha1A = ganA;
alpha2A = ganA;
alpha3A = ganA;
alpha4A = ganA;
alpha5A = ganA;
alpha6A = ganA;
alpha7A = ganA;
alpha8A = ganA;
alpha9A = ganA;
alpha10A = ganA;
alpha11A = ganA;
alpha12A = ganA;

% Sistema Maestro (errores)
z1A = - x(38);
z2A = x(2) - 0 - alpha1A*z1A;
z3A = - x(40);
z4A = x(4) - 0 - alpha3A*z3A;
z5A = - x(42);
% z6A = x(6) - 0 - alpha5A*z5A;
z7A = - x(44);
% z8A = x(8) - 0 - alpha7A*z7A;
z9A = - x(46);
% z10A = x(10) - 0 - alpha9A*z9A;
z11A = - x(48);
% z12A = x(12) - 0 - alpha11A*z11A;

```

```

%% Para trayectoria en circulo
% z6A = x(6) - 0 - alpha5A*z5A;
% z8A = x(8) - 0 - alpha7A*z7A;
% z10A = x(10) + 3*((2*pi)/P)*cos(((2*pi)/P)*t) - alpha9A*z9A;
% z12A = x(12) - 3*((2*pi)/P)*sin(((2*pi)/P)*t) - alpha11A*z11A;

%% Para trayectoria en espiral
% z6A = x(6) - 0 - alpha5A*z5A;
% z8A = x(8) - 0.1 - alpha7A*z7A;
% z10A = x(10) + 2.5*((2*pi)/P)*sin(((2*pi)/P)*t) - alpha9A*z9A;
% z12A = x(12) - 2.5*((2*pi)/P)*cos(((2*pi)/P)*t) - alpha11A*z11A;

%% Para trayectoria en 8
z6A = x(6) - 0 - alpha5A*z5A;
z8A = x(8) - 0 - alpha7A*z7A;
z10A = x(10) + 2.5*((2*pi)/P)*sin(((2*pi)/P)*t) - alpha9A*z9A;
z12A = x(12) - 2.5*((4*pi)/P)*cos(((4*pi)/P)*t) - alpha11A*z11A;

U1A = ((m/(cos(x(1))*cos(x(3)))) * (z7A + g - alpha7A*(z8A + alpha7A*z7A) -
alpha8A*z8A));
U2A = (1/b1)*(z1A - a1*x(4)*x(6) - a2*x(4)*omega - alpha1A*(z2A + alpha1A*z1A) -
alpha2A*z2A);
U3A = (1/b2)*(z3A - a3*x(2)*x(6) - a4*x(2)*omega - alpha3A*(z4A + alpha3A*z3A) -
alpha4A*z4A);
U4A = (1/b3)*(z5A - a5*x(2)*x(4) - alpha5A*(z2A + alpha1A*z1A) - alpha6A*z6A);

UxA = ((m/U1A)*(z9A - alpha9A*(z10A+alpha9A*z9A) - alpha10A*z10A));
UyA = ((m/U1A)*(z11A - alpha11A*(z12A+alpha11A*z11A) - alpha12A*z12A));

uzz=m*(z7A - alpha7A*(z8A + alpha7A*z7A) - alpha8A*z8A);
x3d = atan( (1/(uzz+m*g))*(cos(x5d)*UxA+sin(x5d)*UyA) ); % Pitch deseado
x1d = atan( (cos(x3d)/(uzz+m*g))*(sin(x5d)*UxA-cos(x5d)*UyA) );% Roll deseado

% Calculo de velocidades angulares metodo 1
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
WWWA=inv([b b b b;0 -l*b 0 l*b;-l*b 0 l*b 0;-d d -d d])*([U1A;U2A;U3A;U4A]);
WV1A(it)=WWWA(1);
WV2A(it)=WWWA(2);
WV3A(it)=WWWA(3);
WV4A(it)=WWWA(4);
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% Ecuaciones Diferenciales Sistema Maestro
dx(1) = x(2);
dx(2) = x(4)*x(6)*a1 + x(4)*a2*omega + b1*U2A;
dx(3) = x(4);
dx(4) = x(2)*x(6)*a3 + x(2)*a4*omega + b2*U3A;
dx(5) = x(6);
dx(6) = x(4)*x(2)*a5 + b3*U4A;
dx(7) = x(8);

```

```

dx(8) = -g + ((cos (x(1))*cos (x(3)))/m)* U1A;
dx(9) = x(10);
dx(10) = (UxA/m)*U1A;
dx(11) = x(12);
dx(12) = (UyA/m)*U1A;

%-----
% Sistema Esclavo

% Ganancias Sistema Esclavo
ganB = 10;
alpha1B = ganB;
alpha2B = ganB;
alpha3B = ganB;
alpha4B = ganB;
alpha5B = ganB;
alpha6B = ganB;
alpha7B = ganB;
alpha8B = ganB;
alpha9B = ganB;
alpha10B = ganB;
alpha11B = ganB;
alpha12B = ganB;

% Sistema Esclavo (errores)
z1B = - x(50);
z2B = x(14) - 0 - alpha1B*z1B;
z3B = - x(52);
z4B = x(16) - 0 - alpha3B*z3B;
z5B = - x(54);
z6B = x(18) - 0 - alpha5B*z5B;
z7B = - x(56);
z8B = x(20) - 0 - alpha7B*z7B;
z9B = - x(58);
z10B = x(22) - 0 - alpha9B*z9B;
z11B = - x(60);
z12B = x(24) - 0 - alpha11B*z11B;

U1B = ((m/(cos(x(13))*cos(x(15)))) * (z7B + g - alpha7B*(z8B + alpha7B*z7B) -
alpha8B*z8B));
U2B = (1/b1)*(z1B - a1*x(16)*x(18) - a2*x(16)*omega - alpha1B*(z2B + alpha1B*z1B) -
alpha2B*z2B);
U3B = (1/b2)*(z3B - a3*x(14)*x(18) - a4*x(14)*omega - alpha3B*(z4B + alpha3B*z3B) -
alpha4B*z4B);
U4B = (1/b3)*(z5B - a5*x(14)*x(16) - alpha5B*(z2B + alpha1B*z1B) - alpha6B*z6B);

UxB = ((m/U1B)*(z9B - alpha9B*(z10B+alpha9B*z9B) - alpha10B*z10B));
UyB = ((m/U1B)*(z11B - alpha11B*(z12B+alpha11B*z11B) - alpha12B*z12B));

% Calculo de velocidades angulares metodo 1

```

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
WWWB=inv([b b b b;0 -1*b 0 1*b;-1*b 0 1*b 0;-d d -d d])*([U1B;U2B;U3B;U4B]);
WV1B(it)=WWWB(1);
WV2B(it)=WWWB(2);
WV3B(it)=WWWB(3);
WV4B(it)=WWWB(4);
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

```

```

% Ecuaciones diferenciales de Sistema Esclavo
dx(13) = x(14);
dx(14) = x(16)*x(18)*a1 + x(16)*a2*omega + b1*U2B;
dx(15) = x(16);
dx(16) = x(14)*x(18)*a3 + x(14)*a4*omega + b2*U3B;
dx(17) = x(18);
dx(18) = x(16)*x(14)*a5 + b3*U4B;
dx(19) = x(20);
dx(20) = -g + ((cos (x(13))*cos (x(15)))/m)* U1B;
dx(21) = x(22);
dx(22) = (UxB/m)*U1B;
dx(23) = x(24);
dx(24) = (UyB/m)*U1B;

```

```

%-----

```

```

% Sistema Intruso
% Parametros de Control deseadas
x5dI = 0*(pi/180); % Yaw deseado
% x7dI = f3(nc); % Posicion en Z
% x9dI = f1(nc); % Posicion en X
% x11dI = f2(nc); % Posicion en Y

```

```

% % Seguimiento de trayectoria en circulo
x7dI = 1; % Posicion en Z
x9dI = 3*sin(((2*pi)/P)*t); % Posicion en X
x11dI = 3*cos(((2*pi)/P)*t); % Posicion en Y

```

```

% % Seguimiento de trayectoria en espiral
% x7dI = 0.1*t; % Posicion en Z
% x9dI = 2.5*cos(((2*pi)/P)*t); % Posicion en X
% x11dI = 2.5*sin(((2*pi)/P)*t); % Posicion en Y

```

```

% % Seguimiento de trayectoria en 8
% x7dI = 1; % Posicion en Z
% x9dI = 2.5*cos(((2*pi)/P)*t); % Posicion en X
% x11dI = 2.5*sin(((4*pi)/P)*t); % Posicion en Y

```

```

% Ganancias Sistema Esclavo
ganI = 10;
alpha1I = ganI;
alpha2I = ganI;
alpha3I = ganI;

```

```

alpha4I = ganI;
alpha5I = ganI;
alpha6I = ganI;
alpha7I = ganI;
alpha8I = ganI;
alpha9I = ganI;
alpha10I = ganI;
alpha11I = ganI;
alpha12I = ganI;

```

```

% Sistema Esclavo (errores)

```

```

z1I = - x(62);
z2I = x(26) - 0 - alpha1I*z1I;
z3I = - x(64);
z4I = x(28) - 0 - alpha3I*z3I;
z5I = - x(66);
% z6I = x(30) - 0 - alpha5I*z5I;
z7I = - x(68);
% z8I = x(32) - 0 - alpha7I*z7I;
z9I = - x(70);
% z10I = x(34) - 0 - alpha9I*z9I;
z11I = - x(72);
% z12I = x(36) - 0 - alpha11I*z11I;

```

```

% % Para trayectoria en circulo

```

```

z6I = x(30) - 0 - alpha5I*z5I;
z8I = x(32) - 0 - alpha7I*z7I;
z10I = x(34) + 3*((2*pi)/P)*cos(((2*pi)/P)*t) - alpha9I*z9I;
z12I = x(36) - 3*((2*pi)/P)*sin(((2*pi)/P)*t) - alpha11I*z11I;

```

```

% % Para trayectoria en espiral

```

```

% z6I = x(30) - 0 - alpha5I*z5I;
% z8I = x(32) - 0.1 - alpha7I*z7I;
% z10I = x(34) + 2.5*((2*pi)/P)*sin(((2*pi)/P)*t) - alpha9I*z9I;
% z12I = x(36) - 2.5*((2*pi)/P)*cos(((2*pi)/P)*t) - alpha11I*z11I;

```

```

% % Para trayectoria en 8

```

```

% z6I = x(30) - 0 - alpha5I*z5I;
% z8I = x(32) - 0 - alpha7I*z7I;
% z10I = x(34) + 2.5*((2*pi)/P)*sin(((2*pi)/P)*t) - alpha9I*z9I;
% z12I = x(36) - 2.5*((4*pi)/P)*cos(((4*pi)/P)*t) - alpha11I*z11I;

```

```

U1I = ((m/(cos(x(25))*cos(x(27)))) * (z7I + g - alpha7I*(z8I + alpha7I*z7I) -
alpha8I*z8I));
U2I = (1/b1)*(z1I - a1*x(28)*x(30) - a2*x(28)*omega - alpha1I*(z2I + alpha1I*z1I) -
alpha2I*z2I);
U3I = (1/b2)*(z3I - a3*x(26)*x(30) - a4*x(26)*omega - alpha3I*(z4I + alpha3I*z3I) -
alpha4I*z4I);
U4I = (1/b3)*(z5I - a5*x(26)*x(28) - alpha5I*(z2I + alpha1I*z1I) - alpha6I*z6I);

```

```

UxI = ((m/U1I)*(z9I - alpha9I*(z10I+alpha9I*z9I) - alpha10I*z10I));
UyI = ((m/U1I)*(z11I - alpha11I*(z12I+alpha11I*z11I) - alpha12I*z12I));

uzzI=m*(z7I - alpha7I*(z8I + alpha7I*z7I) - alpha8I*z8I);
x3dI = atan( (1/(uzzI+m*g))*(cos(x5dI)*UxI+sin(x5dI)*UyI) ); % Pitch deseado
x1dI = atan( (cos(x3dI)/(uzzI+m*g))*(sin(x5dI)*UxI-cos(x5dI)*UyI) );% Roll deseado

% Calculo de velocidades angulares metodo 1
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
WWWI=inv([b b b b;0 -1*b 0 1*b;-1*b 0 1*b 0;-d d -d d])*([U1I;U2I;U3I;U4I]);
WV1I(it)=WWWI(1);
WV2I(it)=WWWI(2);
WV3I(it)=WWWI(3);
WV4I(it)=WWWI(4);
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

tiempo(it) = t;
it = 1+it;

% Ecuaciones diferenciales de Sistema Esclavo
dx(25) = x(26);
dx(26) = x(28)*x(30)*a1 + x(28)*a2*omega + b1*U2I;
dx(27) = x(28);
dx(28) = x(26)*x(30)*a3 + x(26)*a4*omega + b2*U3I;
dx(29) = x(30);
dx(30) = x(28)*x(26)*a5 + b3*U4I;
dx(31) = x(32);
dx(32) = -g + ((cos (x(25))*cos (x(27)))/m)* U1I;
dx(33) = x(34);
dx(34) = (UxI/m)*U1I;
dx(35) = x(36);
dx(36) = (UyI/m)*U1I;

%-----
% Acoplamiento Dinamico
% Ganancias
D=100;
gam1=D/10;
gam2=D;
% Fase (1) / Antifase (-1)
BT1 = 1; % Yaw
BT2 = 1; % Pitch
BT3 = 1; % Roll
BT4 = 1; % Z
BT5 = -1; % X
BT6 = 1; % Y
% offset separacion entre maestro / esclavo
Dpitch = 0;
Droll = 0;
Dyaw = 0;

```

```
Dx = 0;
Dy = 0;
Dz = 0;
```

```
% ACOPLAMIENTO DINAMICO - Maestro
```

```
dx(37)= -x(37) + x(38);
dx(38)= -gam1*x(37) - gam2*x(38) - D * (x1d - x(1));
dx(39)= -x(39) + x(40);
dx(40)= -gam1*x(39) - gam2*x(40) - D * (x3d - x(3));
dx(41)= -x(41) + x(42);
dx(42)= -gam1*x(41) - gam2*x(42) - D * (x5d - x(5));
dx(43)= -x(43) + x(44);
dx(44)= -gam1*x(43) - gam2*x(44) - D * (x7d - x(7) - RABz*(sign(x(19)-x(7))) -
RAIz*(sign(x(31)-x(7))) );
dx(45)= -x(45) + x(46);
dx(46)= -gam1*x(45) - gam2*x(46) - D * (x9d - x(9) - RABx*(sign(x(21)-x(9))) -
RAIx*(sign(x(33)-x(9))) );
dx(47)= -x(47) + x(48);
dx(48)= -gam1*x(47) - gam2*x(48) - D * (x11d - x(11) - RABy*(sign(x(23)-x(11))) -
RAIy*(sign(x(35)-x(11))) );
```

```
% ACOPLAMIENTO DINAMICO - Esclavo
```

```
dx(49)= -x(49) + x(50);
dx(50)= -gam1*x(49) - gam2*x(50) - D * (BT1*x(1) - x(13) - Dpitch);
dx(51)= -x(51) + x(52);
dx(52)= -gam1*x(51) - gam2*x(52) - D * (BT2*x(3) - x(15) - Droll);
dx(53)= -x(53) + x(54);
dx(54)= -gam1*x(53) - gam2*x(54) - D * (BT3*x(5) - x(17) - Dyaw);
dx(55)= -x(55) + x(56);
dx(56)= -gam1*x(55) - gam2*x(56) - D * (BT4*x(7) - x(19) - Dz -
RBIz*(sign(x(19)-x(31))) );
dx(57)= -x(57) + x(58);
%% x(58) para trayectorias caoticas y concentricas
% dx(58)= -gam1*x(57) - gam2*x(58) - D * (BT5*x(9) - x(21) - Dx*0.5*x(9) );
%% x(58) para trayectorias convencionales
dx(58)= -gam1*x(57) - gam2*x(58) - D * (BT5*x(9) - x(21) - Dx -
RBIx*(sign(x(21)-x(33))) );
dx(59)= -x(59) + x(60);
%% x(60) para trayectorias caoticas y concentricas
% dx(60)= -gam1*x(59) - gam2*x(60) - D * (BT6*x(11) - x(23) - Dy*0.5*x(11) );
%% x(60) para trayectorias convencionales
dx(60)= -gam1*x(59) - gam2*x(60) - D * (BT6*x(11) - x(23) - Dy -
RBIy*(sign(x(23)-x(35))) );
```

```
% ACOPLAMIENTO DINAMICO - Intruso
```

```
dx(61)= -x(61) + x(62);
dx(62)= -gam1*x(61) - gam2*x(62) - D * (x1dI - x(25) );
dx(63)= -x(63) + x(64);
dx(64)= -gam1*x(63) - gam2*x(64) - D * (x3dI - x(27) );
dx(65)= -x(65) + x(66);
```

```

dx(66)= -gam1*x(65) - gam2*x(66) - D * (x5dI - x(29) );
dx(67)= -x(67) + x(68);
dx(68)= -gam1*x(67) - gam2*x(68) - D * (x7dI - x(31) );
dx(69)= -x(69) + x(70);
dx(70)= -gam1*x(69) - gam2*x(70) - D * (x9dI - x(33) );
dx(71)= -x(71) + x(72);
dx(72)= -gam1*x(71) - gam2*x(72) - D * (x11dI - x(35) );

dx = dx';
end

```

Anexo 2: Código para la simulación del control y acoplamiento de la configuración Maestro-Intermitente Cinco nodos.

```

% Utilizar control por backstepping para hacer formacion de un drone
% Sistemas Dinamicos Maestro & Esclavo
% Se agrego un anticolisiones en el esclavo
% Seguir trayectorias Caoticas Maestro - Esclavo(acoplamiento dinamico)
% 5 Maestros intermitentes siguiendo un punto Caotico
% Se agrego las velocidades de los rotores
% Autor: Johnny Rubio Pecasso
% Version: 7.2
% Diciembre/02/2021
% Actualizacion: Diciembre/04/2021
clear all
clc
close all

```

```

P = 20;
global it;
it = 1;
global contador;
contador = 1;
global nc;
nc=1;
global flags
global Gx;
global Gy;
global Gz;
global PCX;
global PCY;
global PCZ;
PCX = 1;
PCY = 1;
PCZ = 1;
global tiempo;
tiempo = 0;
pas = 0.05;
global x1d;
global x3d;
x1d=0;
x3d=0;
global WWA;
global WV1A;
global WV2A;
global WV3A;
global WV4A;
global WWB;
global WV1B;
global WV2B;
global WV3B;
global WV4B;
global WWC;
global WV1C;

```

```
C1 = y(:,25);
C2 = y(:,27);
C3 = y(:,29);
CZ = y(:,31);
CX = y(:,33);
```

```
CY = y(:,35);
```

```
D1 = y(:,37);
```

```
D2 = y(:,39);
```

```
D3 = y(:,41);
```

```
DZ = y(:,43);
```

```
DX = y(:,45);
```

```
DY = y(:,47);
```

```
E1 = y(:,49);
```

```
E2 = y(:,51);
```

```
E3 = y(:,53);
```

```
EZ = y(:,55);
```

```
EX = y(:,57);
```

```
EY = y(:,59);
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

```
% Sincronizar tiempo de iteracion con salidas
```

```
conteo1 = pas-pas;
```

```
conteo2 = pas;
```

```
j = 1;
```

```
for i=1:length(tiempo)
```

```
    if tiempo(i) > conteo1 && tiempo(i) < conteo2
```

```
        GxV(j) = Gx(i);
```

```
        GyV(j) = Gy(i);
```

```
        GzV(j) = Gz(i);
```

```
        flagsV(j) = flags(i);
```

```
        WVv1A(j) = WV1A(i);
```

```
        WVv2A(j) = WV2A(i);
```

```
        WVv3A(j) = WV3A(i);
```

```
        WVv4A(j) = WV4A(i);
```

```
        WVv1B(j) = WV1B(i);
```

```
        WVv2B(j) = WV2B(1);
```

```
        WVv3B(j) = WV3B(i);
```

```
        WVv4B(j) = WV4B(i);
```

```
        WVv1C(j) = WV1C(i);
```

```
        WVv2C(j) = WV2C(1);
```

```
        WVv3C(j) = WV3C(i);
```

```
        WVv4C(j) = WV4C(i);
```

```
        WVv1D(j) = WV1D(i);
```

```
        WVv2D(j) = WV2D(1);
```

```
        WVv3D(j) = WV3D(i);
```

```
        WVv4D(j) = WV4D(i);
```

```

        WVv1E(j) = WV1E(i);
        WVv2E(j) = WV2E(1);
        WVv3E(j) = WV3E(i);
        WVv4E(j) = WV4E(i);

        conteo1 = conteo1 + pas;
        conteo2 = conteo2 + pas;
        tm(j) = j;
        j = j + 1;
    end
end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Simulacion de drones
pasos = 10;
fig = figure;
set(fig,'position',[0 0 2160 1080]);

subplot(6,2,[1,3,5]);
M1 = Diseno(AX(1),AY(1),AZ(1),A1(1),A2(1),A3(1),6,'r');
M2 = Diseno(BX(1),BY(1),BZ(1),B1(1),B2(1),B3(1),6,'g');
M3 = Diseno(CX(1),CY(1),CZ(1),C1(1),C2(1),C3(1),6,'b');
M4 = Diseno(DX(1),DY(1),DZ(1),D1(1),D2(1),D3(1),6,'y');
M5 = Diseno(EX(1),EY(1),EZ(1),E1(1),E2(1),E3(1),6,'k');
grid on;
axis square; cameratoolbar
axis([-7 7 -7 7 0 2]);

subplot(6,2,[7,9,11]);
M6 = Diseno(AX(1),AY(1),0,A1(1),A2(1),A3(1),6,'r');
M7 = Diseno(BX(1),BY(1),0,B1(1),B2(1),B3(1),6,'g');
M8 = Diseno(CX(1),CY(1),0,C1(1),C2(1),C3(1),6,'b');
M9 = Diseno(DX(1),DY(1),DZ(1),D1(1),D2(1),D3(1),6,'y');
M10 = Diseno(EX(1),EY(1),EZ(1),E1(1),E2(1),E3(1),6,'k');
grid on;
axis([-7 7 -7 7]);

pause(1)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

for i=1:pasos:length(t)
    delete (M1)
    delete (M2)
    delete (M3)
    delete (M4)
    delete (M5)
    delete (M6)
    delete (M7)

```

```

delete (M8)
delete (M9)
delete (M10)

if flagsV(i) == 1
    subplot(6,2,[1,3,5]);
    plot3(GxV(i),GyV(i),GzV(i),'*r','MarkerSize', 7) % Posicion Caotico
    axis([-7 7 -7 7 0 2]);
    hold on;
    grid on;
    M1 = Diseno(AX(i),AY(i),AZ(i),A1(i),A2(i),A3(i),9,'r');
    hold on;
%     plot3(AX(1:i),AY(1:i),AZ(1:i), '-r');
    M2 = Diseno(BX(i),BY(i),BZ(i),B1(i),B2(i),B3(i),6,'g');
    hold on;
%     plot3(BX(1:i),BY(1:i),BZ(1:i), '-g');
    M3 = Diseno(CX(i),CY(i),CZ(i),C1(i),C2(i),C3(i),6,'b');
    hold on;
%     plot3(CX(1:i),CY(1:i),CZ(1:i), '-b');
    M4 = Diseno(DX(i),DY(i),DZ(i),D1(i),D2(i),D3(i),6,'y');
    hold on;
%     plot3(DX(1:i),DY(1:i),DZ(1:i), '-y');
    M5 = Diseno(EX(i),EY(i),EZ(i),E1(i),E2(i),E3(i),6,'k');
    hold on;
%     plot3(EX(1:i),EY(1:i),EZ(1:i), '-k');
    title(' 3D View / Master Chaotic Intermittent - Dynamic Coupling ')
    xlabel('Z Axis')
%     legend('Chaotic Point', 'Master', 'Slave 1', 'Slave 2', 'Slave 3', 'Slave
4')

    subplot(6,2,[2,4]);
    plot3(GxV(i),GyV(i),GzV(i),'*r','MarkerSize', 7) % Posicion Caotico
    axis([-7 7 -7 7]);
    hold on;
    grid on;

    M6 = Diseno(AX(i),AY(i),AZ(i),A1(i),A2(i),A3(i),9,'r');
    hold on;
    plot3(AX(1:i),AY(1:i),AZ(1:i), '-r');
    M7 = Diseno(BX(i),BY(i),BZ(i),B1(i),B2(i),B3(i),6,'g');
    hold on;
    plot3(BX(1:i),BY(1:i),BZ(1:i), '-g');
    M8 = Diseno(CX(i),CY(i),CZ(i),C1(i),C2(i),C3(i),6,'b');
    hold on;
    plot3(CX(1:i),CY(1:i),CZ(1:i), '-b');
    M9 = Diseno(DX(i),DY(i),DZ(i),D1(i),D2(i),D3(i),6,'y');
    hold on;
    plot3(DX(1:i),DY(1:i),DZ(1:i), '-y');
    M10 = Diseno(EX(i),EY(i),EZ(i),E1(i),E2(i),E3(i),6,'k');
    hold on;

```

```

plot3(EX(1:i),EY(1:i),EZ(1:i), '-k');
title(' X & Y View / Master Chaotic Intermittent - Dynamic Coupling ')
xlabel(' X Axis ')
ylabel(' Y Axis ')
legend('Chaotic Point', 'Master', 'Slave 1', 'Slave 2', 'Slave 3', 'Slave
4')

```

```

subplot(6,2,7);
plot(AX(1:i),'r');
hold on;
plot(BX(1:i), 'g');
hold on;
plot(CX(1:i), 'b');
hold on;
plot(DX(1:i), 'y');
hold on;
plot(EX(1:i), 'k');
hold on;
grid on
title(' Temporal X ')
xlabel('Time [s]')

```

```

subplot(6,2,9);
plot(AY(1:i),'r');
hold on;
plot(BY(1:i), 'g');
hold on;
plot(CY(1:i), 'b');
hold on;
plot(DY(1:i), 'y');
hold on;
plot(EY(1:i), 'k');
hold on;
grid on
title(' Temporal Y ')
xlabel('Time [s]')

```

```

subplot(6,2,11);
plot(AZ(1:i),'r');
hold on;
plot(BZ(1:i), 'g');
hold on;
plot(CZ(1:i), 'b');
hold on;
plot(DZ(1:i), 'y');
hold on;
plot(EZ(1:i), 'k');
hold on;
grid on
title(' Temporal Z ')

```

```
xlabel('Time [s]')
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

```
subplot(8,2,10);  
plot(WVv1A(1:i),'r');  
hold on;  
plot(WVv1B(1:i),'g');  
hold on;  
plot(WVv1C(1:i),'b');  
hold on;  
plot(WVv1D(1:i),'y');  
hold on;  
plot(WVv1E(1:i),'k');  
hold on;  
grid on;  
title(' Rotor-1 ')
```

```
subplot(8,2,12);  
plot(WVv2A(1:i),'r');  
hold on;  
plot(WVv2B(1:i),'g');  
hold on;  
plot(WVv2C(1:i),'b');  
hold on;  
plot(WVv2D(1:i),'y');  
hold on;  
plot(WVv2E(1:i),'k');  
grid on;  
title(' Rotor-2 ')
```

```
subplot(8,2,14);  
plot(WVv3A(1:i),'r');  
hold on;  
plot(WVv3B(1:i),'g');  
hold on;  
plot(WVv3C(1:i),'b');  
hold on;  
plot(WVv3D(1:i),'y');  
hold on;  
plot(WVv3E(1:i),'k');  
grid on;  
title(' Rotor-3 ')
```

```
subplot(8,2,16);  
plot(WVv4A(1:i),'r');  
hold on;  
plot(WVv4B(1:i),'g');  
hold on;  
plot(WVv4C(1:i),'b');  
hold on;
```

```

plot(WVv4D(1:i),'y');
hold on;
plot(WVv4E(1:i),'k');
grid on;
title(' Rotor-4 ')
xlabel('Time [s]')
ylabel('Velocity [rpm]')
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

elseif flagsV(i) == 2
    subplot(6,2,[1,3,5]);
    plot3(GxV(i),GyV(i),GzV(i),'*r','MarkerSize', 7) % Posicion Caotico
    axis([-7 7 -7 7 0 2]);
    hold on;
    grid on;

    M1 = Diseno(AX(i),AY(i),AZ(i),A1(i),A2(i),A3(i),6,'r');
    hold on;
%    plot3(AX(1:i),AY(1:i),AZ(1:i), '-r');
    M2 = Diseno(BX(i),BY(i),BZ(i),B1(i),B2(i),B3(i),9,'g');
    hold on;
%    plot3(BX(1:i),BY(1:i),BZ(1:i), '-g');
    M3 = Diseno(CX(i),CY(i),CZ(i),C1(i),C2(i),C3(i),6,'b');
    hold on;
%    plot3(CX(1:i),CY(1:i),CZ(1:i), '-b');
    M4 = Diseno(DX(i),DY(i),DZ(i),D1(i),D2(i),D3(i),6,'y');
    hold on;
%    plot3(DX(1:i),DY(1:i),DZ(1:i), '-y');
    M5 = Diseno(EX(i),EY(i),EZ(i),E1(i),E2(i),E3(i),6,'k');
    hold on;
%    plot3(EX(1:i),EY(1:i),EZ(1:i), '-k');
    title(' X & Y View / Master Chaotic Intermittent - Dynamic Coupling ')
    xlabel(' X Axis ')
    ylabel(' Y Axis ')
%    legend('Chaotic Point', 'Slave 1', 'Master', 'Slave 2', 'Slave 3', 'Slave
4')

    subplot(6,2,[2,4]);
    plot3(GxV(i),GyV(i),GzV(i),'*r','MarkerSize', 7) % Posicion Caotico
    axis([-7 7 -7 7]);
    hold on;
    grid on;

    M6 = Diseno(AX(i),AY(i),AZ(i),A1(i),A2(i),A3(i),6,'r');
    hold on;
    plot3(AX(1:i),AY(1:i),AZ(1:i), '-r');
    M7 = Diseno(BX(i),BY(i),BZ(i),B1(i),B2(i),B3(i),9,'g');
    hold on;
    plot3(BX(1:i),BY(1:i),BZ(1:i), '-g');
    M8 = Diseno(CX(i),CY(i),CZ(i),C1(i),C2(i),C3(i),6,'b');

```

```

hold on;
plot3(CX(1:i),CY(1:i),CZ(1:i), '-b');
M9 = Diseno(DX(i),DY(i),DZ(i),D1(i),D2(i),D3(i),6,'y');
hold on;
plot3(DX(1:i),DY(1:i),DZ(1:i), '-y');
M10 = Diseno(EX(i),EY(i),EZ(i),E1(i),E2(i),E3(i),6,'k');
hold on;
plot3(EX(1:i),EY(1:i),EZ(1:i), '-k');
title(' X & Y View / Master Chaotic Intermittent - Dynamic Coupling ')
xlabel(' X Axis ')
ylabel(' Y Axis ')
legend('Chaotic Point', 'Slave 1', 'Master', 'Slave 2', 'Slave 3', 'Slave

```

4')

```

subplot(6,2,7);
plot(AX(1:i),'r');
hold on;
plot(BX(1:i), 'g');
hold on;
plot(CX(1:i), 'b');
hold on;
plot(DX(1:i), 'y');
hold on;
plot(EX(1:i), 'k');
hold on;
grid on
title(' Temporal X ')
xlabel('Time [s]')

```

```

subplot(6,2,9);
plot(AY(1:i),'r');
hold on;
plot(BY(1:i), 'g');
hold on;
plot(CY(1:i), 'b');
hold on;
plot(DY(1:i), 'y');
hold on;
plot(EY(1:i), 'k');
hold on;
grid on
title(' Temporal Y ')
xlabel('Time [s]')

```

```

subplot(6,2,11);
plot(AZ(1:i),'r');
hold on;
plot(BZ(1:i), 'g');
hold on;
plot(CZ(1:i), 'b');

```

```

hold on;
plot(DZ(1:i), 'y');
hold on;
plot(EZ(1:i), 'k');
hold on;
grid on
title(' Temporal Z ')
xlabel(' Time [s]')

```

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

```

```

subplot(8,2,10);
plot(WVv1A(1:i), 'r');
hold on;
plot(WVv1B(1:i), 'g');
hold on;
plot(WVv1C(1:i), 'b');
hold on;
plot(WVv1D(1:i), 'y');
hold on;
plot(WVv1E(1:i), 'k');
hold on;
grid on;
title(' Rotor-1 ')

```

```

subplot(8,2,12);
plot(WVv2A(1:i), 'r');
hold on;
plot(WVv2B(1:i), 'g');
hold on;
plot(WVv2C(1:i), 'b');
hold on;
plot(WVv2D(1:i), 'y');
hold on;
plot(WVv2E(1:i), 'k');
grid on;
title(' Rotor-2 ')

```

```

subplot(8,2,14);
plot(WVv3A(1:i), 'r');
hold on;
plot(WVv3B(1:i), 'g');
hold on;
plot(WVv3C(1:i), 'b');
hold on;
plot(WVv3D(1:i), 'y');
hold on;
plot(WVv3E(1:i), 'k');
grid on;
title(' Rotor-3 ')

```

```

subplot(8,2,16);
plot(WVv4A(1:i),'r');
hold on;
plot(WVv4B(1:i),'g');
hold on;
plot(WVv4C(1:i),'b');
hold on;
plot(WVv4D(1:i),'y');
hold on;
plot(WVv4E(1:i),'k');
grid on;
title(' Rotor-4 ')
xlabel('Time [s]')
ylabel('Velocity [rpm]')
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

```

```

elseif flagsV(i) == 3

```

```

    subplot(6,2,[1,3,5]);
    plot3(GxV(i),GyV(i),GzV(i),'*r','MarkerSize', 7) % Posicion Caotico
    axis([-7 7 -7 7 0 2]);
    hold on;
    grid on;

```

```

    M1 = Diseno(AX(i),AY(i),AZ(i),A1(i),A2(i),A3(i),6,'r');
    hold on;

```

```

%     plot3(AX(1:i),AY(1:i),AZ(1:i), '-r');

```

```

    M2 = Diseno(BX(i),BY(i),BZ(i),B1(i),B2(i),B3(i),6,'g');
    hold on;

```

```

%     plot3(BX(1:i),BY(1:i),BZ(1:i), '-g');

```

```

    M3 = Diseno(CX(i),CY(i),CZ(i),C1(i),C2(i),C3(i),9,'b');
    hold on;

```

```

%     plot3(CX(1:i),CY(1:i),CZ(1:i), '-b');

```

```

    M4 = Diseno(DX(i),DY(i),DZ(i),D1(i),D2(i),D3(i),6,'y');
    hold on;

```

```

%     plot3(DX(1:i),DY(1:i),DZ(1:i), '-y');

```

```

    M5 = Diseno(EX(i),EY(i),EZ(i),E1(i),E2(i),E3(i),6,'k');
    hold on;

```

```

%     plot3(EX(1:i),EY(1:i),EZ(1:i), '-k');

```

```

    title(' X & Y View / Master Chaotic Intermittent - Dynamic Coupling ')

```

```

    xlabel(' X Axis ')

```

```

    ylabel(' Y Axis ')

```

```

%     legend('Chaotic Point', 'Slave 1', 'Slave 2', 'Master', 'Slave 3', 'Slave
4')

```

```

subplot(6,2,[2,4]);
plot3(GxV(i),GyV(i),GzV(i),'*r','MarkerSize', 7) % Posicion Caotico
axis([-7 7 -7 7]);
hold on;
grid on;

```

```

M6 = Diseno(AX(i),AY(i),AZ(i),A1(i),A2(i),A3(i),6,'r');
hold on;
plot3(AX(1:i),AY(1:i),AZ(1:i), '-r');
M7 = Diseno(BX(i),BY(i),BZ(i),B1(i),B2(i),B3(i),6,'g');
hold on;
plot3(BX(1:i),BY(1:i),BZ(1:i), '-g');
M8 = Diseno(CX(i),CY(i),CZ(i),C1(i),C2(i),C3(i),9,'b');
hold on;
plot3(CX(1:i),CY(1:i),CZ(1:i), '-b');
M9 = Diseno(DX(i),DY(i),DZ(i),D1(i),D2(i),D3(i),6,'y');
hold on;
plot3(DX(1:i),DY(1:i),DZ(1:i), '-y');
M10 = Diseno(EX(i),EY(i),EZ(i),E1(i),E2(i),E3(i),6,'k');
hold on;
plot3(EX(1:i),EY(1:i),EZ(1:i), '-k');
title(' X & Y View / Master Chaotic Intermittent - Dynamic Coupling ')
xlabel(' X Axis ')
ylabel(' Y Axis ')
legend('Chaotic Point', 'Slave 1', 'Slave 2', 'Master', 'Slave 3', 'Slave

```

4')

```

subplot(6,2,7);
plot(AX(1:i),'r');
hold on;
plot(BX(1:i), 'g');
hold on;
plot(CX(1:i), 'b');
hold on;
plot(DX(1:i), 'y');
hold on;
plot(EX(1:i), 'k');
hold on;
grid on
title(' Temporal X ')
xlabel('Time [s]')

```

```

subplot(6,2,9);
plot(AY(1:i),'r');
hold on;
plot(BY(1:i), 'g');
hold on;
plot(CY(1:i), 'b');
hold on;
plot(DY(1:i), 'y');
hold on;
plot(EY(1:i), 'k');
hold on;
grid on
title(' Temporal Y ')
xlabel('Time [s]')

```

```

subplot(6,2,11);
plot(AZ(1:i),'r');
hold on;
plot(BZ(1:i), 'g');
hold on;
plot(CZ(1:i), 'b');
hold on;
plot(DZ(1:i), 'y');
hold on;
plot(EZ(1:i), 'k');
hold on;
grid on
title(' Temporal Z ')
xlabel('Time [s]')

```

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

```

```

subplot(8,2,10);
plot(WVv1A(1:i),'r');
hold on;
plot(WVv1B(1:i),'g');
hold on;
plot(WVv1C(1:i),'b');
hold on;
plot(WVv1D(1:i),'y');
hold on;
plot(WVv1E(1:i),'k');
hold on;
grid on;
title(' Rotor-1 ')

```

```

subplot(8,2,12);
plot(WVv2A(1:i),'r');
hold on;
plot(WVv2B(1:i),'g');
hold on;
plot(WVv2C(1:i),'b');
hold on;
plot(WVv2D(1:i),'y');
hold on;
plot(WVv2E(1:i),'k');
grid on;
title(' Rotor-2 ')

```

```

subplot(8,2,14);
plot(WVv3A(1:i),'r');
hold on;
plot(WVv3B(1:i),'g');
hold on;
plot(WVv3C(1:i),'b');

```

```

hold on;
plot(WVv3D(1:i), 'y');
hold on;
plot(WVv3E(1:i), 'k');
grid on;
title(' Rotor-3 ')

subplot(8,2,16);
plot(WVv4A(1:i), 'r');
hold on;
plot(WVv4B(1:i), 'g');
hold on;
plot(WVv4C(1:i), 'b');
hold on;
plot(WVv4D(1:i), 'y');
hold on;
plot(WVv4E(1:i), 'k');
grid on;
title(' Rotor-4 ')
xlabel('Time [s]')
ylabel('Velocity [rpm]')
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

```

```

elseif flagsV(i) == 4
    subplot(6,2,[1,3,5]);
    plot3(GxV(i),GyV(i),GzV(i), '*r', 'MarkerSize', 7) % Posicion Caotico
    axis([-7 7 -7 7 0 2]);
    hold on;
    grid on;

    M1 = Diseno(AX(i),AY(i),AZ(i),A1(i),A2(i),A3(i),6, 'r');
    hold on;
%     plot3(AX(1:i),AY(1:i),AZ(1:i), '-r');
    M2 = Diseno(BX(i),BY(i),BZ(i),B1(i),B2(i),B3(i),6, 'g');
    hold on;
%     plot3(BX(1:i),BY(1:i),BZ(1:i), '-g');
    M3 = Diseno(CX(i),CY(i),CZ(i),C1(i),C2(i),C3(i),6, 'b');
    hold on;
%     plot3(CX(1:i),CY(1:i),CZ(1:i), '-b');
    M4 = Diseno(DX(i),DY(i),DZ(i),D1(i),D2(i),D3(i),9, 'y');
    hold on;
%     plot3(DX(1:i),DY(1:i),DZ(1:i), '-y');
    M5 = Diseno(EX(i),EY(i),EZ(i),E1(i),E2(i),E3(i),6, 'k');
    hold on;
%     plot3(EX(1:i),EY(1:i),EZ(1:i), '-k');
    title(' X & Y View / Master Chaotic Intermittent - Dynamic Coupling ')
    xlabel(' X Axis ')
    ylabel(' Y Axis ')
%     legend('Chaotic Point', 'Slave 1', 'Slave 2', 'Slave 3', 'Master', 'Slave
4')

```

```

subplot(6,2,[2,4]);
plot3(GxV(i),GyV(i),GzV(i),'*r','MarkerSize', 7) % Posicion Caotico
axis([-7 7 -7 7]);
hold on;
grid on;

M6 = Diseno(AX(i),AY(i),AZ(i),A1(i),A2(i),A3(i),6,'r');
hold on;
plot3(AX(1:i),AY(1:i),AZ(1:i), '-r');
M7 = Diseno(BX(i),BY(i),BZ(i),B1(i),B2(i),B3(i),6,'g');
hold on;
plot3(BX(1:i),BY(1:i),BZ(1:i), '-g');
M8 = Diseno(CX(i),CY(i),CZ(i),C1(i),C2(i),C3(i),6,'b');
hold on;
plot3(CX(1:i),CY(1:i),CZ(1:i), '-b');
M9 = Diseno(DX(i),DY(i),DZ(i),D1(i),D2(i),D3(i),9,'y');
hold on;
plot3(DX(1:i),DY(1:i),DZ(1:i), '-y');
M10 = Diseno(EX(i),EY(i),EZ(i),E1(i),E2(i),E3(i),6,'k');
hold on;
plot3(EX(1:i),EY(1:i),EZ(1:i), '-k');
title(' X & Y View / Master Chaotic Intermittent - Dynamic Coupling ')
xlabel(' X Axis ')
ylabel(' Y Axis ')
legend('Chaotic Point', 'Slave 1', 'Slave 2', 'Slave 3', 'Master', 'Slave

```

4')

```

subplot(6,2,7);
plot(AX(1:i),'r');
hold on;
plot(BX(1:i), 'g');
hold on;
plot(CX(1:i), 'b');
hold on;
plot(DX(1:i), 'y');
hold on;
plot(EX(1:i), 'k');
hold on;
grid on;
title(' Temporal X ')
xlabel('Time [s]')

```

```

subplot(6,2,9);
plot(AY(1:i),'r');
hold on;
plot(BY(1:i), 'g');
hold on;
plot(CY(1:i), 'b');
hold on;

```

```

plot(DY(1:i), 'y');
hold on;
plot(EY(1:i), 'k');
hold on;
grid on
title(' Temporal Y ')
xlabel('Time [s]')

```

```

subplot(6,2,11);
plot(AZ(1:i), 'r');
hold on;
plot(BZ(1:i), 'g');
hold on;
plot(CZ(1:i), 'b');
hold on;
plot(DZ(1:i), 'y');
hold on;
plot(EZ(1:i), 'k');
hold on;
grid on
title(' Temporal Z ')
xlabel('Time [s]')

```

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

```

```

subplot(8,2,10);
plot(WVv1A(1:i), 'r');
hold on;
plot(WVv1B(1:i), 'g');
hold on;
plot(WVv1C(1:i), 'b');
hold on;
plot(WVv1D(1:i), 'y');
hold on;
plot(WVv1E(1:i), 'k');
hold on;
grid on;
title(' Rotor-1 ')

```

```

subplot(8,2,12);
plot(WVv2A(1:i), 'r');
hold on;
plot(WVv2B(1:i), 'g');
hold on;
plot(WVv2C(1:i), 'b');
hold on;
plot(WVv2D(1:i), 'y');
hold on;
plot(WVv2E(1:i), 'k');
grid on;
title(' Rotor-2 ')

```

```

subplot(8,2,14);
plot(WVv3A(1:i),'r');
hold on;
plot(WVv3B(1:i),'g');
hold on;
plot(WVv3C(1:i),'b');
hold on;
plot(WVv3D(1:i),'y');
hold on;
plot(WVv3E(1:i),'k');
grid on;
title(' Rotor-3 ')

```

```

subplot(8,2,16);
plot(WVv4A(1:i),'r');
hold on;
plot(WVv4B(1:i),'g');
hold on;
plot(WVv4C(1:i),'b');
hold on;
plot(WVv4D(1:i),'y');
hold on;
plot(WVv4E(1:i),'k');
grid on;
title(' Rotor-4 ')
xlabel('Time [s]')
ylabel('Velocity [rpm]')

```

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

```

```

elseif flagsV(i) == 5
    subplot(6,2,[1,3,5]);
    plot3(GxV(i),GyV(i),GzV(i),'*r','MarkerSize', 7) % Posicion Caotico
    axis([-7 7 -7 7 0 2]);
    hold on;
    grid on;

    M1 = Diseno(AX(i),AY(i),AZ(i),A1(i),A2(i),A3(i),6,'r');
    hold on;
%     plot3(AX(1:i),AY(1:i),AZ(1:i), '-r');
    M2 = Diseno(BX(i),BY(i),BZ(i),B1(i),B2(i),B3(i),6,'g');
    hold on;
%     plot3(BX(1:i),BY(1:i),BZ(1:i), '-g');
    M3 = Diseno(CX(i),CY(i),CZ(i),C1(i),C2(i),C3(i),6,'b');
    hold on;
%     plot3(CX(1:i),CY(1:i),CZ(1:i), '-b');
    M4 = Diseno(DX(i),DY(i),DZ(i),D1(i),D2(i),D3(i),6,'y');
    hold on;
%     plot3(DX(1:i),DY(1:i),DZ(1:i), '-y');
    M5 = Diseno(EX(i),EY(i),EZ(i),E1(i),E2(i),E3(i),9,'k');

```

```

hold on;
% plot3(EX(1:i),EY(1:i),EZ(1:i), '-k');
title(' X & Y View / Master Chaotic Intermittent - Dynamic Coupling ')
xlabel(' X Axis ')
ylabel(' Y Axis ')
% legend('Chaotic Point', 'Slave 1', 'Slave 2', 'Slave 3', 'Slave 4',
'Master')

subplot(6,2,[2,4]);
plot3(GxV(i),GyV(i),GzV(i),'*r','MarkerSize', 7) % Posicion Caotico
axis([-7 7 -7 7]);
hold on;
grid on;

M6 = Diseno(AX(i),AY(i),AZ(i),A1(i),A2(i),A3(i),6,'r');
hold on;
plot3(AX(1:i),AY(1:i),AZ(1:i), '-r');
M7 = Diseno(BX(i),BY(i),BZ(i),B1(i),B2(i),B3(i),6,'g');
hold on;
plot3(BX(1:i),BY(1:i),BZ(1:i), '-g');
M8 = Diseno(CX(i),CY(i),CZ(i),C1(i),C2(i),C3(i),6,'b');
hold on;
plot3(CX(1:i),CY(1:i),CZ(1:i), '-b');
M9 = Diseno(DX(i),DY(i),DZ(i),D1(i),D2(i),D3(i),6,'y');
hold on;
plot3(DX(1:i),DY(1:i),DZ(1:i), '-y');
M10 = Diseno(EX(i),EY(i),EZ(i),E1(i),E2(i),E3(i),9,'k');
hold on;
plot3(EX(1:i),EY(1:i),EZ(1:i), '-k');
title(' X & Y View / Master Chaotic Intermittent - Dynamic Coupling ')
xlabel(' X Axis ')
ylabel(' Y Axis ')
legend('Chaotic Point', 'Slave 1', 'Slave 2', 'Slave 3', 'Slave 4',
'Master')

subplot(6,2,7);
plot(AX(1:i),'r');
hold on;
plot(BX(1:i), 'g');
hold on;
plot(CX(1:i), 'b');
hold on;
plot(DX(1:i), 'y');
hold on;
plot(EX(1:i), 'k');
hold on;
grid on
title(' Temporal X ')
xlabel('Time [s]')

```

```

subplot(6,2,9);
plot(AY(1:i),'r');
hold on;
plot(BY(1:i), 'g');
hold on;
plot(CY(1:i), 'b');
hold on;
plot(DY(1:i), 'y');
hold on;
plot(EY(1:i), 'k');
hold on;
grid on
title(' Temporal Y ')
xlabel('Time [s]')

```

```

subplot(6,2,11);
plot(AZ(1:i),'r');
hold on;
plot(BZ(1:i), 'g');
hold on;
plot(CZ(1:i), 'b');
hold on;
plot(DZ(1:i), 'y');
hold on;
plot(EZ(1:i), 'k');
hold on;
grid on
title(' Temporal Z ')
xlabel('Time [s]')

```

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

```

```

subplot(8,2,10);
plot(WVv1A(1:i),'r');
hold on;
plot(WVv1B(1:i),'g');
hold on;
plot(WVv1C(1:i),'b');
hold on;
plot(WVv1D(1:i),'y');
hold on;
plot(WVv1E(1:i),'k');
hold on;
grid on;
title(' Rotor-1 ')

```

```

subplot(8,2,12);
plot(WVv2A(1:i),'r');
hold on;
plot(WVv2B(1:i),'g');
hold on;

```

```

plot(WVv2C(1:i),'b');
hold on;
plot(WVv2D(1:i),'y');
hold on;
plot(WVv2E(1:i),'k');
grid on;
title(' Rotor-2 ')

```

```

subplot(8,2,14);
plot(WVv3A(1:i),'r');
hold on;
plot(WVv3B(1:i),'g');
hold on;
plot(WVv3C(1:i),'b');
hold on;
plot(WVv3D(1:i),'y');
hold on;
plot(WVv3E(1:i),'k');
grid on;
title(' Rotor-3 ')

```

```

subplot(8,2,16);
plot(WVv4A(1:i),'r');
hold on;
plot(WVv4B(1:i),'g');
hold on;
plot(WVv4C(1:i),'b');
hold on;
plot(WVv4D(1:i),'y');
hold on;
plot(WVv4E(1:i),'k');
grid on;
title(' Rotor-4 ')
xlabel('Time [s]')
ylabel('Velocity [rpm]')

```

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

```

```

end

```

```

pause(1e-24)

```

```

end

```

```

function dx=SistemaControl5nodosSim(t,x)
% Modelo de la funcion de control de cuadricoptero
% Parametros fisicos
b = 3.13e-5; % Thrust factor
d = 7.5e-7; % Drag factor
% Inercia
Ix = 7.5e-3;
Iy = 7.5e-3;
Iz = 1.3e-2;
Jr = 6e-5; % inercia motor

```

```
l = .23; % lever
b1 = l / Ix;
b2 = l / Iy;
b3 = l / Iz;
g = 9.81; % Gravedad
m = .650; % masa
P = 20; % Periodo
```

```
global xxC
global yC
global zC
global f1
global f2
global f3
global nc
global flag
global flags
global Gx
global Gy
global Gz
global PCX
global PCY
global PCZ
global it
global tiempo
global x1d;
global x3d;
global tiempo;
global WWWA;
global WV1A;
global WV2A;
global WV3A;
global WV4A;
global WWWB;
global WV1B;
global WV2B;
global WV3B;
global WV4B;
global WWWC;
global WV1C;
global WV2C;
global WV3C;
global WV4C;
global WWWD;
global WV1D;
global WV2D;
global WV3D;
global WV4D;
global WWWE;
global WV1E;
```

```

global WV2E;
global WV3E;
global WV4E;

% Velocidades de los rotores
O1 = 15000; % Rotor 1
O2 = 15000; % Rotor 2
O3 = 15000; % Rotor 3
O4 = 15000; % Rotor 4
omega = O2 + O4 - O1 - O3;

% Variables
a1=(Iy-Iz)/Ix;
a2=(-Jr/Ix);
a3=(Iz-Ix)/Iy;
a4=(-Jr/Iy);
a5=(Ix-Iy)/Iz;

%-----
% Evacion de colisiones
% Evacion entre A-B
Rm = 0.15; % Coeficiente de anticollision
Um = 0.1; % Umbral de colision
DAB = sqrt( (x(19)-x(7))^2 + (x(21)-x(9))^2 + (x(23)-x(11))^2 ); % Distancia entre Drones
% Desicion para distancia entre drones
if DAB <= Um
    RABz = Rm;
    RABx = Rm;
    RABy = Rm;
else
    RABz = 0;
    RABx = 0;
    RABy = 0;
end

% Evacion entre A-C
DAC = sqrt( (x(31)-x(7))^2 + (x(33)-x(9))^2 + (x(33)-x(11))^2 ); % Distancia entre Drones
% Desicion para distancia entre drones
if DAC <= Um
    RACz = Rm;
    RACx = Rm;
    RACy = Rm;
else
    RACz = 0;
    RACx = 0;
    RACy = 0;
end

```

```

% Evacion entre A-D
DAD = sqrt( (x(43)-x(7))^2 + (x(45)-x(9))^2 + (x(47)-x(11))^2 ); % Distancia entre Drones
% Desicion para distancia entre drones
if DAD <= Um
    RADz = Rm;
    RADx = Rm;
    RADy = Rm;
else
    RADz = 0;
    RADx = 0;
    RADy = 0;
end

% Evacion entre A-E
DAE = sqrt( (x(55)-x(7))^2 + (x(57)-x(9))^2 + (x(59)-x(11))^2 ); % Distancia entre Drones
% Desicion para distancia entre drones
if DAE <= Um
    RAEz = Rm;
    RAEy = Rm;
    RAEy = Rm;
else
    RAEz = 0;
    RAEy = 0;
    RAEy = 0;
end

% Evacion entre B-C
DBC = sqrt( (x(31)-x(19))^2 + (x(33)-x(21))^2 + (x(35)-x(23))^2 ); % Distancia entre Drones
% Desicion para distancia entre drones
if DBC <= Um
    RBCz = Rm;
    RBCx = Rm;
    RBCy = Rm;
else
    RBCz = 0;
    RBCx = 0;
    RBCy = 0;
end

% Evacion entre B-D
DBD = sqrt( (x(43)-x(19))^2 + (x(45)-x(21))^2 + (x(47)-x(23))^2 ); % Distancia entre Drones
% Desicion para distancia entre drones
if DBD <= Um
    RBDz = Rm;
    RBDx = Rm;
    RBDy = Rm;

```

```

else
    RBDz = 0;
    RBDx = 0;
    RBDy = 0;
end

% Evacion entre B-E
DBE = sqrt( (x(55)-x(19))^2 + (x(57)-x(21))^2 + (x(59)-x(23))^2 ); % Distancia entre Drones
% Desicion para distancia entre drones
if DBE <= Um
    RBEz = Rm;
    RBEx = Rm;
    RBEy = Rm;
else
    RBEz = 0;
    RBEx = 0;
    RBEy = 0;
end

% Evacion entre C-D
DCD = sqrt( (x(43)-x(31))^2 + (x(45)-x(33))^2 + (x(47)-x(35))^2 ); % Distancia entre Drones
% Desicion para distancia entre drones
if DCD <= Um
    RCDz = Rm;
    RCDx = Rm;
    RCDy = Rm;
else
    RCDz = 0;
    RCDx = 0;
    RCDy = 0;
end

% Evacion entre C-E
DCE = sqrt( (x(55)-x(31))^2 + (x(57)-x(33))^2 + (x(59)-x(35))^2 ); % Distancia entre Drones
% Desicion para distancia entre drones
if DCE <= Um
    RCEz = Rm;
    RCEx = Rm;
    RCEy = Rm;
else
    RCEz = 0;
    RCEx = 0;
    RCEy = 0;
end

% Evacion entre D-E
DDE = sqrt( (x(55)-x(43))^2 + (x(57)-x(45))^2 + (x(59)-x(47))^2 ); % Distancia entre

```

```

Drones
% Desicion para distancia entre drones
if DDE <= Um
    RDEz = Rm;
    RDEx = Rm;
    RDEy = Rm;
else
    RDEz = 0;
    RDEx = 0;
    RDEy = 0;
end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%-----
% Ruta caotica para Maestro
xxC(1)=1;
yC(1)=1;
zC(1)=1;
f1(1)=xxC(1);
f2(1)=yC(1);
f3(1)=zC(1);
tao=0.01;
aC=10;
cC=8/3;
bC=28;
caos = 0.02; % Variable para aumentar las rutas deseadas caoticas

if abs(f1(nc)-PCX)< caos && abs(f2(nc)-PCY)< caos && abs(f3(nc)-PCZ)< caos
    % Atractor Caotico de Lorenz
    xxC(nc+1)=xxC(nc)+tao*(aC.*(yC(nc)-xxC(nc)));
    yC(nc+1)=yC(nc)+tao*(xxC(nc)*(bC-zC(nc))-yC(nc));
    zC(nc+1)=zC(nc)+tao*(yC(nc)*xxC(nc)-cC*zC(nc));

    nc=nc+1;

    f1(nc)=floor(((xxC(nc)*1000)-floor(xxC(nc)*1000))*6)-2.5;
    f2(nc)=floor(((yC(nc)*1000)-floor(yC(nc)*1000))*6)-2.5;
    f3(nc)=floor(((zC(nc)*1000)-floor(zC(nc)*1000))*0.5)+.75;

%     f1(nc)=(rand(1,1)*6)-2.5;
%     f2(nc)=(rand(1,1)*6)-2.5;
%     f3(nc)=(rand(1,1)*0.5)+0.75;

%-----
% Desicion para maestro intermitente
DistMA = sqrt((f1(nc)-x(9))^2 + (f2(nc)-x(11))^2 + (f3(nc)-x(7))^2);
DistMB = sqrt((f1(nc)-x(21))^2 + (f2(nc)-x(23))^2 + (f3(nc)-x(19))^2);
DistMC = sqrt((f1(nc)-x(33))^2 + (f2(nc)-x(35))^2 + (f3(nc)-x(31))^2);
DistMD = sqrt((f1(nc)-x(45))^2 + (f2(nc)-x(47))^2 + (f3(nc)-x(43))^2);
DistME = sqrt((f1(nc)-x(57))^2 + (f2(nc)-x(59))^2 + (f3(nc)-x(55))^2);

```

```

if DistMA < DistMB && DistMA < DistMC && DistMA < DistMD && DistMA < DistME
    flag = 1;
elseif DistMB < DistMA && DistMB < DistMC && DistMB < DistMD && DistMB < DistME
    flag = 2;
elseif DistMC < DistMA && DistMC < DistMB && DistMC < DistMD && DistMC < DistME
    flag = 3;
elseif DistMD < DistMA && DistMD < DistMB && DistMD < DistMC && DistMD < DistME
    flag = 4;
elseif DistME < DistMA && DistME < DistMB && DistME < DistMC && DistME < DistMD
    flag = 5;
end

```

end

```

% %%%%%%%%%%%
% Parametros de Control deseadas
x5d = 0*(pi/180); % Yaw deseado
x7d = f3(nc); % Posicion en Z
x9d = f1(nc); % Posicion en X
x11d = f2(nc); % Posicion en Y
% %%%%%%%%%%%

```

```

%-----

```

```

% Sistema Intermitente A

```

```

% Ganancias

```

```

ganA = 10;
alpha1A = ganA;
alpha2A = ganA;
alpha3A = ganA;
alpha4A = ganA;
alpha5A = ganA;
alpha6A = ganA;
alpha7A = ganA;
alpha8A = ganA;
alpha9A = ganA;
alpha10A = ganA;
alpha11A = ganA;
alpha12A = ganA;

```

```

% Sistema A (errores)

```

```

z1A = - x(62);
z2A = x(2) - 0 - alpha1A*z1A;
z3A = - x(64);
z4A = x(4) - 0 - alpha3A*z3A;
z5A = - x(66);
z6A = x(6) - 0 - alpha5A*z5A;
z7A = - x(68);
z8A = x(8) - 0 - alpha7A*z7A;
z9A = - x(70);

```

```

z10A = x(10) - 0 - alpha9A*z9A;
z11A = - x(72);
z12A = x(12) - 0 - alpha11A*z11A;

U1A = ((m/(cos(x(1))*cos(x(3)))) * (z7A + g - alpha7A*(z8A + alpha7A*z7A) -
alpha8A*z8A));
U2A = (1/b1)*(z1A - a1*x(4)*x(6) - a2*x(4)*omega - alpha1A*(z2A + alpha1A*z1A) -
alpha2A*z2A);
U3A = (1/b2)*(z3A - a3*x(2)*x(6) - a4*x(2)*omega - alpha3A*(z4A + alpha3A*z3A) -
alpha4A*z4A);
U4A = (1/b3)*(z5A - a5*x(2)*x(4) - alpha5A*(z2A + alpha1A*z1A) - alpha6A*z6A);

UxA = ((m/U1A)*(z9A - alpha9A*(z10A+alpha9A*z9A) - alpha10A*z10A));
UyA = ((m/U1A)*(z11A - alpha11A*(z12A+alpha11A*z11A) - alpha12A*z12A));

uzz=m*(z7A - alpha7A*(z8A + alpha7A*z7A) - alpha8A*z8A);
x3d = atan( (1/(uzz+m*g))*(cos(x5d)*UxA+sin(x5d)*UyA) ); % Pitch deseado
x1d = atan( (cos(x3d) /(uzz+m*g))*(sin(x5d)*UxA-cos(x5d)*UyA) );% Roll deseado

% Calculo de velocidades angulares metodo 1
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
WWWA=inv([b b b b;0 -1*b 0 1*b;-1*b 0 1*b 0;-d d -d d])*([U1A;U2A;U3A;U4A]);
WV1A(it)=WWWA(1);
WV2A(it)=WWWA(2);
WV3A(it)=WWWA(3);
WV4A(it)=WWWA(4);
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% Ecuaciones Diferenciales Sistema A
dx(1) = x(2);
dx(2) = x(4)*x(6)*a1 + x(4)*a2*omega + b1*U2A;
dx(3) = x(4);
dx(4) = x(2)*x(6)*a3 + x(2)*a4*omega + b2*U3A;
dx(5) = x(6);
dx(6) = x(4)*x(2)*a5 + b3*U4A;
dx(7) = x(8);
dx(8) = -g + ((cos (x(1))*cos (x(3)))/m)* U1A;
dx(9) = x(10);
dx(10) = (UxA/m)*U1A;
dx(11) = x(12);
dx(12) = (UyA/m)*U1A;

%-----
% Sistema Intermitente B

% Ganancias Sistema B
ganB = 10;
alpha1B = ganB;
alpha2B = ganB;
alpha3B = ganB;

```

```

alpha4B = ganB;
alpha5B = ganB;
alpha6B = ganB;
alpha7B = ganB;
alpha8B = ganB;
alpha9B = ganB;
alpha10B = ganB;
alpha11B = ganB;
alpha12B = ganB;

% Sistema B (errores)
z1B = - x(74);
z2B = x(14) - 0 - alpha1B*z1B;
z3B = - x(76);
z4B = x(16) - 0 - alpha3B*z3B;
z5B = - x(78);
z6B = x(18) - 0 - alpha5B*z5B;
z7B = - x(80);
z8B = x(20) - 0 - alpha7B*z7B;
z9B = - x(82);
z10B = x(22) - 0 - alpha9B*z9B;
z11B = - x(84);
z12B = x(24) - 0 - alpha11B*z11B;

U1B = ((m/(cos(x(13))*cos(x(15)))) * (z7B + g - alpha7B*(z8B + alpha7B*z7B) -
alpha8B*z8B));
U2B = (1/b1)*(z1B - a1*x(16)*x(18) - a2*x(16)*omega - alpha1B*(z2B + alpha1B*z1B) -
alpha2B*z2B);
U3B = (1/b2)*(z3B - a3*x(14)*x(18) - a4*x(14)*omega - alpha3B*(z4B + alpha3B*z3B) -
alpha4B*z4B);
U4B = (1/b3)*(z5B - a5*x(14)*x(16) - alpha5B*(z2B + alpha1B*z1B) - alpha6B*z6B);

UxB = ((m/U1B)*(z9B - alpha9B*(z10B+alpha9B*z9B) - alpha10B*z10B));
UyB = ((m/U1B)*(z11B - alpha11B*(z12B+alpha11B*z11B) - alpha12B*z12B));

% Calculo de velocidades angulares metodo 1
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
WWWB=inv([b b b b;0 -l*b 0 l*b;-l*b 0 l*b 0;-d d -d d])*([U1B;U2B;U3B;U4B]);
WV1B(it)=WWWB(1);
WV2B(it)=WWWB(2);
WV3B(it)=WWWB(3);
WV4B(it)=WWWB(4);
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% Ecuaciones diferenciales de Sistema B
dx(13) = x(14);
dx(14) = x(16)*x(18)*a1 + x(16)*a2*omega + b1*U2B;
dx(15) = x(16);
dx(16) = x(14)*x(18)*a3 + x(14)*a4*omega + b2*U3B;
dx(17) = x(18);

```

```

dx(18) = x(16)*x(14)*a5 + b3*U4B;
dx(19) = x(20);
dx(20) = -g + ((cos (x(13))*cos (x(15)))/m)* U1B;
dx(21) = x(22);
dx(22) = (UxB/m)*U1B;
dx(23) = x(24);
dx(24) = (UyB/m)*U1B;

%-----
% Sistema Intermitente C

% Ganancias Sistema C
ganC = 10;
alpha1C = ganC;
alpha2C = ganC;
alpha3C = ganC;
alpha4C = ganC;
alpha5C = ganC;
alpha6C = ganC;
alpha7C = ganC;
alpha8C = ganC;
alpha9C = ganC;
alpha10C = ganC;
alpha11C = ganC;
alpha12C = ganC;

% Sistema C Dinamico (errores)
z1C = - x(86);
z2C = x(26) - 0 - alpha1C*z1C;
z3C = - x(88);
z4C = x(28) - 0 - alpha3C*z3C;
z5C = - x(90);
z6C = x(30) - 0 - alpha5C*z5C;
z7C = - x(92);
z8C = x(32) - 0 - alpha7C*z7C;
z9C = - x(94);
z10C = x(34) - 0 - alpha9C*z9C;
z11C = - x(96);
z12C = x(36) - 0 - alpha11C*z11C;

U1C = ((m/(cos(x(25))*cos(x(27)))) * (z7C + g - alpha7C*(z8C + alpha7C*z7C) -
alpha8C*z8C));
U2C = (1/b1)*(z1C - a1*x(28)*x(30) - a2*x(28)*omega - alpha1C*(z2C + alpha1C*z1C) -
alpha2C*z2C);
U3C = (1/b2)*(z3C - a3*x(26)*x(30) - a4*x(26)*omega - alpha3C*(z4C + alpha3C*z3C) -
alpha4C*z4C);
U4C = (1/b3)*(z5C - a5*x(26)*x(28) - alpha5C*(z2C + alpha1C*z1C) - alpha6C*z6C);

UxC = ((m/U1C)*(z9C - alpha9C*(z10C+alpha9C*z9C) - alpha10C*z10C));
UyC = ((m/U1C)*(z11C - alpha11C*(z12C+alpha11C*z11C) - alpha12C*z12C));

```

```
% Calculo de velocidades angulares metodo 1
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
WWWC=inv([b b b b;0 -l*b 0 l*b;-l*b 0 l*b 0;-d d -d d])*([U1C;U2C;U3C;U4C]);
WV1C(it)=WWWC(1);
WV2C(it)=WWWC(2);
WV3C(it)=WWWC(3);
WV4C(it)=WWWC(4);
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

```
% Ecuaciones diferenciales de Sistema C
dx(25) = x(26);
dx(26) = x(28)*x(30)*a1 + x(28)*a2*omega + b1*U2C;
dx(27) = x(28);
dx(28) = x(26)*x(30)*a3 + x(26)*a4*omega + b2*U3C;
dx(29) = x(30);
dx(30) = x(28)*x(26)*a5 + b3*U4C;
dx(31) = x(32);
dx(32) = -g + ((cos (x(25))*cos (x(27)))/m)* U1C;
dx(33) = x(34);
dx(34) = (UxC/m)*U1C;
dx(35) = x(36);
dx(36) = (UyC/m)*U1C;
```

```
%-----
% Sistema Intermitente D
```

```
% Ganancias Sistema D
ganD = 10;
alpha1D = ganD;
alpha2D = ganD;
alpha3D = ganD;
alpha4D = ganD;
alpha5D = ganD;
alpha6D = ganD;
alpha7D = ganD;
alpha8D = ganD;
alpha9D = ganD;
alpha10D = ganD;
alpha11D = ganD;
alpha12D = ganD;
```

```
% Sistema D Dinamico (errores)
z1D = - x(98);
z2D = x(38) - 0 - alpha1D*z1D;
z3D = - x(100);
z4D = x(40) - 0 - alpha3D*z3D;
z5D = - x(102);
z6D = x(42) - 0 - alpha5D*z5D;
z7D = - x(104);
```

```

z8D = x(44) - 0 - alpha7D*z7D;
z9D = - x(106);
z10D = x(46) - 0 - alpha9D*z9D;
z11D = - x(108);
z12D = x(48) - 0 - alpha11D*z11D;

```

```

U1D = ((m/(cos(x(37))*cos(x(39)))) * (z7D + g - alpha7D*(z8D + alpha7D*z7D) -
alpha8D*z8D));
U2D = (1/b1)*(z1D - a1*x(40)*x(42) - a2*x(40)*omega - alpha1D*(z2D + alpha1D*z1D) -
alpha2D*z2D);
U3D = (1/b2)*(z3D - a3*x(38)*x(42) - a4*x(38)*omega - alpha3D*(z4D + alpha3D*z3D) -
alpha4D*z4D);
U4D = (1/b3)*(z5D - a5*x(38)*x(40) - alpha5D*(z2D + alpha1D*z1D) - alpha6D*z6D);

UxD = ((m/U1D)*(z9D - alpha9D*(z10D+alpha9D*z9D) - alpha10D*z10D));
UyD = ((m/U1D)*(z11D - alpha11D*(z12D+alpha11D*z11D) - alpha12D*z12D));

```

```

% Calculo de velocidades angulares metodo 1
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
WWWD=inv([b b b b;0 -1*b 0 1*b;-1*b 0 1*b 0;-d d -d d])*([U1D;U2D;U3D;U4D]);
WV1D(it)=WWWD(1);
WV2D(it)=WWWD(2);
WV3D(it)=WWWD(3);
WV4D(it)=WWWD(4);
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

```

```

% Ecuaciones diferenciales de Sistema D
dx(37) = x(38);
dx(38) = x(40)*x(42)*a1 + x(40)*a2*omega + b1*U2D;
dx(39) = x(40);
dx(40) = x(38)*x(42)*a3 + x(38)*a4*omega + b2*U3D;
dx(41) = x(42);
dx(42) = x(40)*x(38)*a5 + b3*U4D;
dx(43) = x(44);
dx(44) = -g + ((cos (x(37))*cos (x(39)))/m)* U1D;
dx(45) = x(46);
dx(46) = (UxD/m)*U1D;
dx(47) = x(48);
dx(48) = (UyD/m)*U1D;

```

```

%-----
% Sistema Intermitente E

```

```

% Ganancias Sistema Dinamico
ganE = 10;
alpha1E = ganE;
alpha2E = ganE;
alpha3E = ganE;
alpha4E = ganE;
alpha5E = ganE;

```

```

alpha6E = ganE;
alpha7E = ganE;
alpha8E = ganE;
alpha9E = ganE;
alpha10E = ganE;
alpha11E = ganE;
alpha12E = ganE;

% Sistema E Dinamico (errores)
z1E = - x(110);
z2E = x(50) - 0 - alpha1E*z1E;
z3E = - x(112);
z4E = x(52) - 0 - alpha3E*z3E;
z5E = - x(114);
z6E = x(54) - 0 - alpha5E*z5E;
z7E = - x(116);
z8E = x(56) - 0 - alpha7E*z7E;
z9E = - x(118);
z10E = x(58) - 0 - alpha9E*z9E;
z11E = - x(120);
z12E = x(60) - 0 - alpha11E*z11E;

U1E = ((m/(cos(x(49))*cos(x(51)))) * (z7E + g - alpha7E*(z8E + alpha7E*z7E) -
alpha8E*z8E));
U2E = (1/b1)*(z1E - a1*x(52)*x(54) - a2*x(52)*omega - alpha1E*(z2E + alpha1E*z1E) -
alpha2E*z2E);
U3E = (1/b2)*(z3E - a3*x(50)*x(54) - a4*x(50)*omega - alpha3E*(z4E + alpha3E*z3E) -
alpha4E*z4E);
U4E = (1/b3)*(z5E - a5*x(50)*x(52) - alpha5E*(z2E + alpha1E*z1E) - alpha6E*z6E);

UxE = ((m/U1E)*(z9E - alpha9E*(z10E+alpha9E*z9E) - alpha10E*z10E));
UyE = ((m/U1E)*(z11E - alpha11E*(z12E+alpha11E*z11E) - alpha12E*z12E));

% Calculo de velocidades angulares metodo 1
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
WWWE=inv([b b b b;0 -l*b 0 l*b;-l*b 0 l*b 0;-d d -d d])*([U1E;U2E;U3E;U4E]);
WV1E(it)=WWWE(1);
WV2E(it)=WWWE(2);
WV3E(it)=WWWE(3);
WV4E(it)=WWWE(4);
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% Ecuaciones diferenciales de Sistema E
dx(49) = x(50);
dx(50) = x(52)*x(54)*a1 + x(54)*a2*omega + b1*U2E;
dx(51) = x(52);
dx(52) = x(50)*x(54)*a3 + x(50)*a4*omega + b2*U3E;
dx(53) = x(54);
dx(54) = x(52)*x(50)*a5 + b3*U4E;
dx(55) = x(56);

```

```

dx(56) = -g + ((cos (x(49))*cos (x(51)))/m)* U1E;
dx(57) = x(58);
dx(58) = (UxE/m)*U1E;
dx(59) = x(60);
dx(60) = (UyE/m)*U1E;

%-----
% Acoplamiento Dinámico
% Ganancias
D=100;
gam1=D/10;
gam2=D;
% Fase (1) / Antifase (-1)
BT1 = 1; % Yaw
BT2 = 1; % Pitch
BT3 = 1; % Roll
BT4 = 1; % Z
BT5 = 1; % X
BT6 = 1; % Y
% offset separacion entre maestro / esclavo
Dpitch = 0;
Droll = 0;
Dyaw = 0;
Dx = 1;
Dy = 1;
Dz = 0;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
if flag == 1

PCX = x(9);
PCY = x(11);
PCZ = x(7);

Gx(it) = x9d;
Gy(it) = x11d;
Gz(it) = x7d;

flags(it) = flag;

tiempo(it) = t;
it = it + 1;

% ACOPLAMIENTO DINAMICO INTERMITENTE A - Maestro
dx(61)= -x(61) + x(62);
dx(62)= -gam1*x(61) - gam2*x(62) - D * (x1d - x(1));
dx(63)= -x(63) + x(64);
dx(64)= -gam1*x(63) - gam2*x(64) - D * (x3d - x(3));
dx(65)= -x(65) + x(66);
dx(66)= -gam1*x(65) - gam2*x(66) - D * (x5d - x(5));

```

```

dx(67)= -x(67) + x(68);
dx(68)= -gam1*x(67) - gam2*x(68) - D * (x7d - x(7));
dx(69)= -x(69) + x(70);
dx(70)= -gam1*x(69) - gam2*x(70) - D * (x9d - x(9));
dx(71)= -x(71) + x(72);
dx(72)= -gam1*x(71) - gam2*x(72) - D * (x11d - x(11));

% ACOPLAMIENTO DINAMICO INTERMITENTE B
dx(73)= -x(73) + x(74);
dx(74)= -gam1*x(73) - gam2*x(74) - D * (BT1*x(1) - x(13) - Dpitch);
dx(75)= -x(75) + x(76);
dx(76)= -gam1*x(75) - gam2*x(76) - D * (BT2*x(3) - x(15) - Droll);
dx(77)= -x(77) + x(78);
dx(78)= -gam1*x(77) - gam2*x(78) - D * (BT3*x(5) - x(17) - Dyaw);
dx(79)= -x(79) + x(80);
dx(80)= -gam1*x(79) - gam2*x(80) - D * (BT4*x(7) - x(19) - Dz -
RABz*(sign(x(19)-x(7))) - RBCz*(sign(x(31)-x(19))) - RBDz*(sign(x(43)-x(19))) -
RBEz*(sign(x(55)-x(19))) );
dx(81)= -x(81) + x(82);
dx(82)= -gam1*x(81) - gam2*x(82) - D * (BT5*x(9) - x(21) - Dx*-0.5*x(9) -
RABx*(sign(x(21)-x(9))) - RBCx*(sign(x(33)-x(21))) - RBDx*(sign(x(45)-x(21))) -
RBEy*(sign(x(57)-x(21))) );
dx(83)= -x(83) + x(84);
dx(84)= -gam1*x(83) - gam2*x(84) - D * (BT6*x(11) - x(23) - Dy*0*x(11) -
RABx*(sign(x(23)-x(11))) - RBCy*(sign(x(35)-x(23))) - RBDy*(sign(x(47)-x(23))) -
RBEy*(sign(x(59)-x(23))) );

% ACOPLAMIENTO DINAMICO INTERMITENTE C
dx(85)= -x(85) + x(86);
dx(86)= -gam1*x(85) - gam2*x(86) - D * (BT1*x(1) - x(25) - Dpitch);
dx(87)= -x(87) + x(88);
dx(88)= -gam1*x(87) - gam2*x(88) - D * (BT2*x(3) - x(27) - Droll);
dx(89)= -x(89) + x(90);
dx(90)= -gam1*x(89) - gam2*x(90) - D * (BT3*x(5) - x(29) - Dyaw);
dx(91)= -x(91) + x(92);
dx(92)= -gam1*x(91) - gam2*x(92) - D * (BT4*x(7) - x(31) - Dz -
RACz*(sign(x(31)-x(7))) - RBCz*(sign(x(31)-x(19))) - RCDz*(sign(x(43)-x(31))) -
RCEz*(sign(x(55)-x(31))) );
dx(93)= -x(93) + x(94);
dx(94)= -gam1*x(93) - gam2*x(94) - D * (BT5*x(9) - x(33) - Dx*0.5*x(9) -
RACx*(sign(x(33)-x(9))) - RBCx*(sign(x(33)-x(21))) - RCDx*(sign(x(45)-x(33))) -
RCEx*(sign(x(57)-x(33))) );
dx(95)= -x(95) + x(96);
dx(96)= -gam1*x(95) - gam2*x(96) - D * (BT6*x(11)- x(35) - Dy*0*x(11) -
RACy*(sign(x(35)-x(11))) - RBCy*(sign(x(35)-x(23))) - RCDy*(sign(x(47)-x(35))) -
RCEy*(sign(x(59)-x(35))) );

% ACOPLAMIENTO DINAMICO INTERMITENTE D
dx(97)= -x(97) + x(98);
dx(98)= -gam1*x(97) - gam2*x(98) - D * (BT1*x(1) - x(37) - Dpitch);

```

```

dx(99)= -x(99) + x(100);
dx(100)= -gam1*x(99) - gam2*x(100) - D * (BT2*x(3) - x(39) - Droll);
dx(101)= -x(101) + x(102);
dx(102)= -gam1*x(101) - gam2*x(102) - D * (BT3*x(5) - x(41) - Dyaw);
dx(103)= -x(103) + x(104);
dx(104)= -gam1*x(103) - gam2*x(104) - D * (BT4*x(7) - x(43) - Dz -
RADz*(sign(x(43)-x(7))) - RBDz*(sign(x(43)-x(19))) - RCDz*(sign(x(43)-x(31))) -
RDEz*(sign(x(43)-x(55))) );
dx(105)= -x(105) + x(106);
dx(106)= -gam1*x(105) - gam2*x(106) - D * (BT5*x(9) - x(45) - Dx*0*x(9) -
RADx*(sign(x(45)-x(9))) - RBDx*(sign(x(45)-x(21))) - RCDx*(sign(x(45)-x(33))) -
RDEx*(sign(x(45)-x(57)))));
dx(107)= -x(107) + x(108);
dx(108)= -gam1*x(107) - gam2*x(108) - D * (BT6*x(11) - x(47) - Dy*-0.5*x(11) -
RADy*(sign(x(47)-x(11))) - RBDy*(sign(x(47)-x(23))) - RCDy*(sign(x(47)-x(35))) -
RDEy*(sign(x(47)-x(59))) );

```

% ACOPLAMIENTO DINAMICO INTERMITENTE E

```

dx(109)= -x(109) + x(110);
dx(110)= -gam1*x(109) - gam2*x(110) - D * (BT1*x(1) - x(49) - Dpitch);
dx(111)= -x(111) + x(112);
dx(112)= -gam1*x(111) - gam2*x(112) - D * (BT2*x(3) - x(51) - Droll);
dx(113)= -x(113) + x(114);
dx(114)= -gam1*x(113) - gam2*x(114) - D * (BT3*x(5) - x(53) - Dyaw);
dx(115)= -x(115) + x(116);
dx(116)= -gam1*x(115) - gam2*x(116) - D * (BT4*x(7) - x(55) - Dz -
RAEz*(sign(x(55)-x(7))) - RBEz*(sign(x(55)-x(19))) - RCEz*(sign(x(55)-x(31))) -
RDEz*(sign(x(43)-x(55))) );
dx(117)= -x(117) + x(118);
dx(118)= -gam1*x(117) - gam2*x(118) - D * (BT5*x(9) - x(57) - Dx*0*x(9) -
RAEx*(sign(x(57)-x(9))) - RBEx*(sign(x(57)-x(21))) - RCEx*(sign(x(57)-x(33))) -
RDEx*(sign(x(45)-x(57))) );
dx(119)= -x(119) + x(120);
dx(120)= -gam1*x(119) - gam2*x(120) - D * (BT6*x(11) - x(59) - Dy*0.5*x(11) -
RAEy*(sign(x(59)-x(11))) - RBEy*(sign(x(59)-x(23))) - RCEy*(sign(x(59)-x(35))) -
RDEy*(sign(x(47)-x(59))) );

```

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

```

```

elseif flag == 2

```

```

PCX = x(21);
PCY = x(23);
PCZ = x(19);

```

```

Gx(it) = x9d;
Gy(it) = x11d;
Gz(it) = x7d;

```

```

flags(it) = flag;

```

```

tiempo(it) = t;
it = it + 1;

```

```

% ACOPLAMIENTO DINAMICO INTERMITENTE A

```

```

dx(61)= -x(61) + x(62);
dx(62)= -gam1*x(61) - gam2*x(62) - D * (BT1*x(13) - x(1) - Dpitch);
dx(63)= -x(63) + x(64);
dx(64)= -gam1*x(63) - gam2*x(64) - D * (BT2*x(15) - x(3) - Droll);
dx(65)= -x(65) + x(66);
dx(66)= -gam1*x(65) - gam2*x(66) - D * (BT3*x(17) - x(5) - Dyaw);
dx(67)= -x(67) + x(68);
dx(68)= -gam1*x(67) - gam2*x(68) - D * (BT4*x(19) - x(7) - Dz -
RABz*(sign(x(19)-x(7))) - RACz*(sign(x(31)-x(7))) - RADz*(sign(x(43)-x(7))) -
RAEz*(sign(x(55)-x(7))) );
dx(69)= -x(69) + x(70);
dx(70)= -gam1*x(69) - gam2*x(70) - D * (BT5*x(21) - x(9) - Dx*-0.5*x(21) -
RABx*(sign(x(21)-x(9))) - RACx*(sign(x(33)-x(9))) - RADx*(sign(x(45)-x(9))) -
RAEx*(sign(x(57)-x(9))) );
dx(71)= -x(71) + x(72);
dx(72)= -gam1*x(71) - gam2*x(72) - D * (BT6*x(23) - x(11) - Dy*0*x(23) -
RABy*(sign(x(23)-x(11))) - RACy*(sign(x(35)-x(11))) - RADy*(sign(x(47)-x(11))) -
RAEy*(sign(x(59)-x(11))) );

```

```

% ACOPLAMIENTO DINAMICO INTERMITENTE B - Maestro

```

```

dx(73)= -x(73) + x(74);
dx(74)= -gam1*x(73) - gam2*x(74) - D * (x1d - x(13));
dx(75)= -x(75) + x(76);
dx(76)= -gam1*x(75) - gam2*x(76) - D * (x3d - x(15));
dx(77)= -x(77) + x(78);
dx(78)= -gam1*x(77) - gam2*x(78) - D * (x5d - x(17));
dx(79)= -x(79) + x(80);
dx(80)= -gam1*x(79) - gam2*x(80) - D * (x7d - x(19));
dx(81)= -x(81) + x(82);
dx(82)= -gam1*x(81) - gam2*x(82) - D * (x9d - x(21));
dx(83)= -x(83) + x(84);
dx(84)= -gam1*x(83) - gam2*x(84) - D * (x11d - x(23));

```

```

% ACOPLAMIENTO DINAMICO INTERMITENTE C

```

```

dx(85)= -x(85) + x(86);
dx(86)= -gam1*x(85) - gam2*x(86) - D * (BT1*x(13) - x(25) - Dpitch);
dx(87)= -x(87) + x(88);
dx(88)= -gam1*x(87) - gam2*x(88) - D * (BT2*x(15) - x(27) - Droll);
dx(89)= -x(89) + x(90);
dx(90)= -gam1*x(89) - gam2*x(90) - D * (BT3*x(17) - x(29) - Dyaw);
dx(91)= -x(91) + x(92);
dx(92)= -gam1*x(91) - gam2*x(92) - D * (BT4*x(19) - x(31) - Dz -
RACz*(sign(x(31)-x(7))) - RBCz*(sign(x(31)-x(19))) - RCDz*(sign(x(43)-x(31))) -
RCEz*(sign(x(55)-x(31))) );
dx(93)= -x(93) + x(94);
dx(94)= -gam1*x(93) - gam2*x(94) - D * (BT5*x(21) - x(33) - Dx*0.5*x(21) -

```



```
PCX = x(33);
PCY = x(35);
PCZ = x(31);
```

```
Gx(it) = x9d;
Gy(it) = x11d;
Gz(it) = x7d;
```

```
flags(it) = flag;
```

```
tiempo(it) = t;
it = it + 1;
```

```
% ACOPLAMIENTO DINAMICO INTERMITENTE A
```

```
dx(61)= -x(61) + x(62);
dx(62)= -gam1*x(61) - gam2*x(62) - D * (BT1*x(25) - x(1) - Dpitch);
dx(63)= -x(63) + x(64);
dx(64)= -gam1*x(63) - gam2*x(64) - D * (BT2*x(27) - x(3) - Droll);
dx(65)= -x(65) + x(66);
dx(66)= -gam1*x(65) - gam2*x(66) - D * (BT3*x(29) - x(5) - Dyaw);
dx(67)= -x(67) + x(68);
dx(68)= -gam1*x(67) - gam2*x(68) - D * (BT4*x(31) - x(7) - Dz -
RABz*(sign(x(19)-x(7))) - RACz*(sign(x(31)-x(7))) - RADz*(sign(x(43)-x(7))) -
RAEz*(sign(x(55)-x(7))) );
dx(69)= -x(69) + x(70);
dx(70)= -gam1*x(69) - gam2*x(70) - D * (BT5*x(33) - x(9) - Dx*0.5*x(33) -
RABx*(sign(x(21)-x(9))) - RACx*(sign(x(33)-x(9))) - RADx*(sign(x(45)-x(9))) -
RAEx*(sign(x(57)-x(9))) );
dx(71)= -x(71) + x(72);
dx(72)= -gam1*x(71) - gam2*x(72) - D * (BT6*x(35) - x(11) - Dy*0*x(35) -
RABy*(sign(x(23)-x(11))) - RACy*(sign(x(35)-x(11))) - RADy*(sign(x(47)-x(11))) -
RAEy*(sign(x(59)-x(11))) );
```

```
% ACOPLAMIENTO DINAMICO INTERMITENTE B
```

```
dx(73)= -x(73) + x(74);
dx(74)= -gam1*x(73) - gam2*x(74) - D * (BT1*x(25) - x(13) - Dpitch);
dx(75)= -x(75) + x(76);
dx(76)= -gam1*x(75) - gam2*x(76) - D * (BT2*x(27) - x(15) - Droll);
dx(77)= -x(77) + x(78);
dx(78)= -gam1*x(77) - gam2*x(78) - D * (BT3*x(29) - x(17) - Dyaw);
dx(79)= -x(79) + x(80);
dx(80)= -gam1*x(79) - gam2*x(80) - D * (BT4*x(31) - x(19) - Dz -
RABz*(sign(x(19)-x(7))) - RBCz*(sign(x(31)-x(19))) - RBDz*(sign(x(43)-x(19))) -
RBEz*(sign(x(55)-x(19))) );
dx(81)= -x(81) + x(82);
dx(82)= -gam1*x(81) - gam2*x(82) - D * (BT5*x(33) - x(21) - Dx*0.5*x(33) -
RABx*(sign(x(21)-x(9))) - RBCx*(sign(x(33)-x(21))) - RBDx*(sign(x(45)-x(21))) -
RBEx*(sign(x(57)-x(21))) );
dx(83)= -x(83) + x(84);
dx(84)= -gam1*x(83) - gam2*x(84) - D * (BT6*x(35) - x(23) - Dy*0*x(35) -
```

```
RABy*(sign(x(23)-x(11))) - RBCy*(sign(x(35)-x(23))) - RBDy*(sign(x(47)-x(23))) -
RBEy*(sign(x(59)-x(23))) );
```

```
% ACOPLAMIENTO DINAMICO INTERMITENTE C - Maestro
```

```
dx(85)= -x(85) + x(86);
dx(86)= -gam1*x(85) - gam2*x(86) - D * (x1d - x(25));
dx(87)= -x(87) + x(88);
dx(88)= -gam1*x(87) - gam2*x(88) - D * (x3d - x(27));
dx(89)= -x(89) + x(90);
dx(90)= -gam1*x(89) - gam2*x(90) - D * (x5d - x(29));
dx(91)= -x(91) + x(92);
dx(92)= -gam1*x(91) - gam2*x(92) - D * (x7d - x(31));
dx(93)= -x(93) + x(94);
dx(94)= -gam1*x(93) - gam2*x(94) - D * (x9d - x(33));
dx(95)= -x(95) + x(96);
dx(96)= -gam1*x(95) - gam2*x(96) - D * (x11d - x(35));
```

```
% ACOPLAMIENTO DINAMICO INTERMITENTE D
```

```
dx(97)= -x(97) + x(98);
dx(98)= -gam1*x(97) - gam2*x(98) - D * (BT1*x(25) - x(37) - Dpitch);
dx(99)= -x(99) + x(100);
dx(100)= -gam1*x(99) - gam2*x(100) - D * (BT2*x(27) - x(39) - Droll);
dx(101)= -x(101) + x(102);
dx(102)= -gam1*x(101) - gam2*x(102) - D * (BT3*x(29) - x(41) - Dyaw);
dx(103)= -x(103) + x(104);
dx(104)= -gam1*x(103) - gam2*x(104) - D * (BT4*x(31) - x(43) - Dz -
RADz*(sign(x(43)-x(7))) - RBDz*(sign(x(43)-x(19))) - RCDz*(sign(x(43)-x(31))) -
RDEz*(sign(x(43)-x(55))) );
dx(105)= -x(105) + x(106);
dx(106)= -gam1*x(105) - gam2*x(106) - D * (BT5*x(33) - x(45) - Dx*0*x(33) -
RADx*(sign(x(45)-x(9))) - RBDx*(sign(x(45)-x(21))) - RCDx*(sign(x(45)-x(33))) -
RDEx*(sign(x(45)-x(57))) );
dx(107)= -x(107) + x(108);
dx(108)= -gam1*x(107) - gam2*x(108) - D * (BT6*x(35) - x(47) - Dy*-0.5*x(35) -
RADy*(sign(x(47)-x(11))) - RBDy*(sign(x(47)-x(23))) - RCDy*(sign(x(47)-x(35))) -
RDEy*(sign(x(47)-x(59))) );
```

```
% ACOPLAMIENTO DINAMICO INTERMITENTE E
```

```
dx(109)= -x(109) + x(110);
dx(110)= -gam1*x(109) - gam2*x(110) - D * (BT1*x(25) - x(49) - Dpitch);
dx(111)= -x(111) + x(112);
dx(112)= -gam1*x(111) - gam2*x(112) - D * (BT2*x(27) - x(51) - Droll);
dx(113)= -x(113) + x(114);
dx(114)= -gam1*x(113) - gam2*x(114) - D * (BT3*x(29) - x(53) - Dyaw);
dx(115)= -x(115) + x(116);
dx(116)= -gam1*x(115) - gam2*x(116) - D * (BT4*x(31) - x(55) - Dz -
RAEz*(sign(x(55)-x(7))) - RBEz*(sign(x(55)-x(19))) - RCEz*(sign(x(55)-x(31))) -
RDEz*(sign(x(43)-x(55))) );
dx(117)= -x(117) + x(118);
dx(118)= -gam1*x(117) - gam2*x(118) - D * (BT5*x(33) - x(57) - Dx*0*x(33) -
```

```

RAEx*(sign(x(57)-x(9))) - RBEEx*(sign(x(57)-x(21))) - RCEEx*(sign(x(57)-x(33))) -
RDEEx*(sign(x(45)-x(57))) );
dx(119)= -x(119) + x(120);
dx(120)= -gam1*x(119) - gam2*x(120) - D * (BT6*x(35) - x(59) - Dy*0.5*x(35) -
RAEy*(sign(x(59)-x(11))) - RBEy*(sign(x(59)-x(23))) - RCEy*(sign(x(59)-x(35))) -
RDEy*(sign(x(47)-x(59))) );

```

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

```

```

elseif flag == 4

```

```

PCX = x(45);
PCY = x(47);
PCZ = x(43);

```

```

Gx(it) = x9d;
Gy(it) = x11d;
Gz(it) = x7d;

```

```

flags(it) = flag;

```

```

tiempo(it) = t;
it = it + 1;

```

```

% ACOPLAMIENTO DINAMICO INTERMITENTE A

```

```

dx(61)= -x(61) + x(62);
dx(62)= -gam1*x(61) - gam2*x(62) - D * (BT1*x(37) - x(1) - Dpitch);
dx(63)= -x(63) + x(64);
dx(64)= -gam1*x(63) - gam2*x(64) - D * (BT2*x(39) - x(3) - Droll);
dx(65)= -x(65) + x(66);
dx(66)= -gam1*x(65) - gam2*x(66) - D * (BT3*x(41) - x(5) - Dyaw);
dx(67)= -x(67) + x(68);
dx(68)= -gam1*x(67) - gam2*x(68) - D * (BT4*x(43) - x(7) - Dz -
RABz*(sign(x(19)-x(7))) - RACz*(sign(x(31)-x(7))) - RADz*(sign(x(43)-x(7))) -
RAEz*(sign(x(55)-x(7))) );
dx(69)= -x(69) + x(70);
dx(70)= -gam1*x(69) - gam2*x(70) - D * (BT5*x(45) - x(9) - Dx*-0.5*x(45) -
RABx*(sign(x(21)-x(9))) - RACx*(sign(x(33)-x(9))) - RADx*(sign(x(45)-x(9))) -
RAEx*(sign(x(57)-x(9))) );
dx(71)= -x(71) + x(72);
dx(72)= -gam1*x(71) - gam2*x(72) - D * (BT6*x(47) - x(11) - Dy*0*x(47) -
RABy*(sign(x(23)-x(11))) - RACy*(sign(x(35)-x(11))) - RADy*(sign(x(47)-x(11))) -
RAEy*(sign(x(59)-x(11))) );

```

```

% ACOPLAMIENTO DINAMICO INTERMITENTE B

```

```

dx(73)= -x(73) + x(74);
dx(74)= -gam1*x(73) - gam2*x(74) - D * (BT2*x(37) - x(13) - Dpitch);
dx(75)= -x(75) + x(76);
dx(76)= -gam1*x(75) - gam2*x(76) - D * (BT2*x(39) - x(15) - Droll);
dx(77)= -x(77) + x(78);
dx(78)= -gam1*x(77) - gam2*x(78) - D * (BT3*x(41) - x(17) - Dyaw);

```

```

dx(79)= -x(79) + x(80);
dx(80)= -gam1*x(79) - gam2*x(80) - D * (BT4*x(43) - x(19) - Dz -
RABz*(sign(x(19)-x(7))) - RBCz*(sign(x(31)-x(19))) - RBDz*(sign(x(43)-x(19))) -
RBEz*(sign(x(55)-x(19))) );
dx(81)= -x(81) + x(82);
dx(82)= -gam1*x(81) - gam2*x(82) - D * (BT5*x(45) - x(21) - Dx*0.5*x(45) -
RABx*(sign(x(21)-x(9))) - RBCx*(sign(x(33)-x(21))) - RBDx*(sign(x(45)-x(21))) -
RBEy*(sign(x(57)-x(21))) );
dx(83)= -x(83) + x(84);
dx(84)= -gam1*x(83) - gam2*x(84) - D * (BT6*x(47)- x(23) - Dy*0*x(47) -
RABx*(sign(x(23)-x(11))) - RBCy*(sign(x(35)-x(23))) - RBDy*(sign(x(47)-x(23))) -
RBEy*(sign(x(59)-x(23))) );

```

% ACOPLAMIENTO DINAMICO INTERMITENTE C

```

dx(85)= -x(85) + x(86);
dx(86)= -gam1*x(85) - gam2*x(86) - D * (BT1*x(37) - x(25) - Dpitch);
dx(87)= -x(87) + x(88);
dx(88)= -gam1*x(87) - gam2*x(88) - D * (BT2*x(39) - x(27) - Droll);
dx(89)= -x(89) + x(90);
dx(90)= -gam1*x(89) - gam2*x(90) - D * (BT3*x(41) - x(29) - Dyaw);
dx(91)= -x(91) + x(92);
dx(92)= -gam1*x(91) - gam2*x(92) - D * (BT4*x(43) - x(31) - Dz -
RACz*(sign(x(31)-x(7))) - RBCz*(sign(x(31)-x(19))) - RCDz*(sign(x(43)-x(31))) -
RCEz*(sign(x(55)-x(31))) );
dx(93)= -x(93) + x(94);
dx(94)= -gam1*x(93) - gam2*x(94) - D * (BT5*x(45) - x(33) - Dx*0*x(45) -
RACx*(sign(x(33)-x(9))) - RBCx*(sign(x(33)-x(21))) - RCDx*(sign(x(45)-x(33))) -
RCEy*(sign(x(57)-x(33))) );
dx(95)= -x(95) + x(96);
dx(96)= -gam1*x(95) - gam2*x(96) - D * (BT6*x(47) - x(35) - Dy*-0.5*x(47) -
RACy*(sign(x(35)-x(11))) - RBCy*(sign(x(35)-x(23))) - RCDy*(sign(x(47)-x(35))) -
RCEy*(sign(x(59)-x(35))) );

```

% ACOPLAMIENTO DINAMICO INTERMITENTE D - Maestro

```

dx(97)= -x(97) + x(98);
dx(98)= -gam1*x(97) - gam2*x(98) - D * (x1d - x(37));
dx(99)= -x(99) + x(100);
dx(100)= -gam1*x(99) - gam2*x(100) - D * (x3d - x(39));
dx(101)= -x(101) + x(102);
dx(102)= -gam1*x(101) - gam2*x(102) - D * (x5d - x(41));
dx(103)= -x(103) + x(104);
dx(104)= -gam1*x(103) - gam2*x(104) - D * (x7d - x(43));
dx(105)= -x(105) + x(106);
dx(106)= -gam1*x(105) - gam2*x(106) - D * (x9d - x(45));
dx(107)= -x(107) + x(108);
dx(108)= -gam1*x(107) - gam2*x(108) - D * (x11d - x(47));

```

% ACOPLAMIENTO DINAMICO INTERMITENTE E

```

dx(109)= -x(109) + x(110);
dx(110)= -gam1*x(109) - gam2*x(110) - D * (BT1*x(37) - x(49) - Dpitch);

```

```

dx(111)= -x(111) + x(112);
dx(112)= -gam1*x(111) - gam2*x(112) - D * (BT2*x(39) - x(51) - Droll);
dx(113)= -x(113) + x(114);
dx(114)= -gam1*x(113) - gam2*x(114) - D * (BT3*x(41) - x(53) - Dyaw);
dx(115)= -x(115) + x(116);
dx(116)= -gam1*x(115) - gam2*x(116) - D * (BT4*x(43) - x(55) - Dz -
RAEz*(sign(x(55)-x(7))) - RBEz*(sign(x(55)-x(19))) - RCEz*(sign(x(55)-x(31))) -
RDEz*(sign(x(43)-x(55))) );
dx(117)= -x(117) + x(118);
dx(118)= -gam1*x(117) - gam2*x(118) - D * (BT5*x(45) - x(57) - Dx*0*x(45) -
RAEx*(sign(x(57)-x(9))) - RBEx*(sign(x(57)-x(21))) - RCEx*(sign(x(57)-x(33))) -
RDEx*(sign(x(45)-x(57))) );
dx(119)= -x(119) + x(120);
dx(120)= -gam1*x(119) - gam2*x(120) - D * (BT6*x(47) - x(59) - Dy*0.5*x(47) -
RAEy*(sign(x(59)-x(11))) - RBEy*(sign(x(59)-x(23))) - RCEy*(sign(x(59)-x(35))) -
RDEy*(sign(x(47)-x(59))) );

```

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
elseif flag == 5

```

```

PCX = x(57);
PCY = x(59);
PCZ = x(55);

```

```

Gx(it) = x9d;
Gy(it) = x11d;
Gz(it) = x7d;

```

```

flags(it) = flag;

```

```

tiempo(it) = t;
it = it + 1;

```

```

% ACOPLAMIENTO DINAMICO INTERMITENTE A

```

```

dx(61)= -x(61) + x(62);
dx(62)= -gam1*x(61) - gam2*x(62) - D * (BT1*x(49) - x(1) - Dpitch);
dx(63)= -x(63) + x(64);
dx(64)= -gam1*x(63) - gam2*x(64) - D * (BT2*x(51) - x(3) - Droll);
dx(65)= -x(65) + x(66);
dx(66)= -gam1*x(65) - gam2*x(66) - D * (BT3*x(53) - x(5) - Dyaw);
dx(67)= -x(67) + x(68);
dx(68)= -gam1*x(67) - gam2*x(68) - D * (BT4*x(55) - x(7) - Dz -
RABz*(sign(x(19)-x(7))) - RACz*(sign(x(31)-x(7))) - RADz*(sign(x(43)-x(7))) -
RAEz*(sign(x(55)-x(7))) );
dx(69)= -x(69) + x(70);
dx(70)= -gam1*x(69) - gam2*x(70) - D * (BT5*x(57) - x(9) - Dx*-0.5*x(57) -
RABx*(sign(x(21)-x(9))) - RACx*(sign(x(33)-x(9))) - RADx*(sign(x(45)-x(9))) -
RAEx*(sign(x(57)-x(9))) );
dx(71)= -x(71) + x(72);
dx(72)= -gam1*x(71) - gam2*x(72) - D * (BT6*x(59) - x(11) - Dy*0*x(59) -

```

```
RABy*(sign(x(23)-x(11))) - RACy*(sign(x(35)-x(11))) - RADy*(sign(x(47)-x(11))) -
RAEy*(sign(x(59)-x(11))) );
```

```
% ACOPLAMIENTO DINAMICO INTERMITENTE B
```

```
dx(73)= -x(73) + x(74);
dx(74)= -gam1*x(73) - gam2*x(74) - D * (BT2*x(49) - x(13) - Dpitch);
dx(75)= -x(75) + x(76);
dx(76)= -gam1*x(75) - gam2*x(76) - D * (BT2*x(51) - x(15) - Droll);
dx(77)= -x(77) + x(78);
dx(78)= -gam1*x(77) - gam2*x(78) - D * (BT3*x(53) - x(17) - Dyaw);
dx(79)= -x(79) + x(80);
dx(80)= -gam1*x(79) - gam2*x(80) - D * (BT4*x(55) - x(19) - Dz -
RABz*(sign(x(19)-x(7))) - RBCz*(sign(x(31)-x(19))) - RBDz*(sign(x(43)-x(19))) -
RBEz*(sign(x(55)-x(19))) );
dx(81)= -x(81) + x(82);
dx(82)= -gam1*x(81) - gam2*x(82) - D * (BT5*x(57) - x(21) - Dx*0.5*x(57) -
RABx*(sign(x(21)-x(9))) - RBCx*(sign(x(33)-x(21))) - RBDx*(sign(x(45)-x(21))) -
RBEy*(sign(x(57)-x(21))) );
dx(83)= -x(83) + x(84);
dx(84)= -gam1*x(83) - gam2*x(84) - D * (BT6*x(59) - x(23) - Dy*0*x(59) -
RABx*(sign(x(23)-x(11))) - RBCy*(sign(x(35)-x(23))) - RBDy*(sign(x(47)-x(23))) -
RBEy*(sign(x(59)-x(23))) );
```

```
% ACOPLAMIENTO DINAMICO INTERMITENTE C
```

```
dx(85)= -x(85) + x(86);
dx(86)= -gam1*x(85) - gam2*x(86) - D * (BT1*x(49) - x(25) - Dpitch);
dx(87)= -x(87) + x(88);
dx(88)= -gam1*x(87) - gam2*x(88) - D * (BT2*x(51) - x(27) - Droll);
dx(89)= -x(89) + x(90);
dx(90)= -gam1*x(89) - gam2*x(90) - D * (BT3*x(53) - x(29) - Dyaw);
dx(91)= -x(91) + x(92);
dx(92)= -gam1*x(91) - gam2*x(92) - D * (BT4*x(55) - x(31) - Dz -
RACz*(sign(x(31)-x(7))) - RBCz*(sign(x(31)-x(19))) - RCDz*(sign(x(43)-x(31))) -
RCEz*(sign(x(55)-x(31))) );
dx(93)= -x(93) + x(94);
dx(94)= -gam1*x(93) - gam2*x(94) - D * (BT5*x(57) - x(33) - Dx*0*x(57) -
RACx*(sign(x(33)-x(9))) - RBCx*(sign(x(33)-x(21))) - RCDx*(sign(x(45)-x(33))) -
RCEx*(sign(x(57)-x(33))) );
dx(95)= -x(95) + x(96);
dx(96)= -gam1*x(95) - gam2*x(96) - D * (BT6*x(59) - x(35) - Dy*-0.5*x(59) -
RACy*(sign(x(35)-x(11))) - RBCy*(sign(x(35)-x(23))) - RCDy*(sign(x(47)-x(35))) -
RCEy*(sign(x(59)-x(35))) );
```

```
% ACOPLAMIENTO DINAMICO INTERMITENTE D
```

```
dx(97)= -x(97) + x(98);
dx(98)= -gam1*x(97) - gam2*x(98) - D * (BT1*x(49) - x(37) - Dpitch);
dx(99)= -x(99) + x(100);
dx(100)= -gam1*x(99) - gam2*x(100) - D * (BT2*x(51) - x(39) - Droll);
dx(101)= -x(101) + x(102);
dx(102)= -gam1*x(101) - gam2*x(102) - D * (BT3*x(53) - x(41) - Dyaw);
```

```

dx(103)= -x(103) + x(104);
dx(104)= -gam1*x(103) - gam2*x(104) - D * (BT4*x(55) - x(43) - Dz -
RADz*(sign(x(43)-x(7))) - RBDz*(sign(x(43)-x(19))) - RCDz*(sign(x(43)-x(31))) -
RDEz*(sign(x(43)-x(55))) );
dx(105)= -x(105) + x(106);
dx(106)= -gam1*x(105) - gam2*x(106) - D * (BT5*x(57) - x(45) - Dx*0*x(57) -
RADx*(sign(x(45)-x(9))) - Dx*0.25*x(21) - RBDx*(sign(x(45)-x(21))) -
RCDx*(sign(x(45)-x(33))) - RDEx*(sign(x(45)-x(57))) );
dx(107)= -x(107) + x(108);
dx(108)= -gam1*x(107) - gam2*x(108) - D * (BT6*x(59) - x(47) - Dy*0.5*x(59) -
RADy*(sign(x(47)-x(11))) - RBDy*(sign(x(47)-x(23))) - RCDy*(sign(x(47)-x(35))) -
RDEy*(sign(x(47)-x(59))) );

```

% ACOPLAMIENTO DINAMICO INTERMITENTE E - Maestro

```

dx(109)= -x(109) + x(110);
dx(110)= -gam1*x(109) - gam2*x(110) - D * (x1d - x(49));
dx(111)= -x(111) + x(112);
dx(112)= -gam1*x(111) - gam2*x(112) - D * (x3d - x(51));
dx(113)= -x(113) + x(114);
dx(114)= -gam1*x(113) - gam2*x(114) - D * (x5d - x(53));
dx(115)= -x(115) + x(116);
dx(116)= -gam1*x(115) - gam2*x(116) - D * (x7d - x(55));
dx(117)= -x(117) + x(118);
dx(118)= -gam1*x(117) - gam2*x(118) - D * (x9d - x(57));
dx(119)= -x(119) + x(120);
dx(120)= -gam1*x(119) - gam2*x(120) - D * (x11d - x(59));
end

```

```

dx = dx';
end

```

Anexo 3: Código para la simulación de animaciones.

```

function Diseno= Plot_Drone(dx,dy,dz,ax,ay,az,scale,colorp)

scale=0.01*scale;

az=-pi/2-az;

Rx=[ 1, 0, 0; 0, cos(ax), -sin(ax); 0, sin(ax), cos(ax)];
Ry=[ cos(ay), 0, sin(ay); 0, 1, 0; -sin(ay), 0, cos(ay)];
Rz=[ cos(az), -sin(az), 0; sin(az), cos(az), 0; 0, 0,1];

R=Rx*Ry*Rz;

% cuerpo
cuerpo=[
-20.457 -9.917 0;
-8.524 -3.027 0;
-8.524 3.178 0;
-20.457 10.068 0;
22.032 10.977 0;
8.524 3.178 0;
8.524 -3.027 0;
22.032 -10.826 0]*1/(3.937*3);

zb=0.2;

cuerpo1=[cuerpo(:,1) cuerpo(:,2) cuerpo(:,3)+zb];

tam=1;

for k=1:length(cuerpo(:,1))-1
    sp1=[cuerpo(k,1) cuerpo(k,2) cuerpo(k,3);cuerpo(k+1,1) cuerpo(k+1,2)
cuerpo(k,3); cuerpo(k+1,1) cuerpo(k+1,2) cuerpo1(k,3); cuerpo(k,1) cuerpo(k,2)
cuerpo1(k,3)]*scale*R;
    Diseno(k)=patch(sp1(:,1)+dx,sp1(:,2)+dy,sp1(:,3)+dz,[0.1 0.1 0.1]);
    if k==length(cuerpo(:,1))-1
        sp1=[cuerpo(length(cuerpo),1) cuerpo(length(cuerpo),2) cuerpo(k,3); cuerpo(1,1)
cuerpo(1,2) cuerpo(k,3); cuerpo(1,1) cuerpo(1,2)
cuerpo1(k,3);cuerpo(length(cuerpo),1) cuerpo(length(cuerpo),2)
cuerpo1(k,3)]*scale*R;
        Diseno(k+1)= patch(sp1(:,1)+dx,sp1(:,2)+dy,sp1(:,3)+dz,[0.1 0.1 0.1]);
    end
end

tam=tam+k;
cuerpo=cuerpo*scale*R;
cuerpo1=cuerpo1*scale*R;
Diseno(tam+1)=patch(cuerpo(:,1)+dx,cuerpo(:,2)+dy,cuerpo(:,3)+dz,[0.2 0.2 0.2]);
Diseno(tam+2)=patch(cuerpo1(:,1)+dx,cuerpo1(:,2)+dy,cuerpo1(:,3)+dz,[0.2 0.2 0.2]);

% cubre elices

```

```

a=pi/10;
th=pi/4:a:2*pi+pi/4;

x=1.4142*cos(th)*0.5;
y=1.4142*sin(th)*0.5;
z=0.21*ones(1,length(th));

hp1=[x'+1.7 y'+1 z']*scale*R;
hp2=[x'-1.7 y'+1 z']*scale*R;
hp3=[x'-1.7 y'-1 z']*scale*R;
hp4=[x'+1.7 y'-1 z']*scale*R;

Diseno(tam+3)=patch(hp1(:,1)+dx,hp1(:,2)+dy,hp1(:,3)+dz,[0.1 0.1
0.1], 'FaceAlpha',0.25);
Diseno(tam+4)=patch(hp2(:,1)+dx,hp2(:,2)+dy,hp2(:,3)+dz,[0.1 0.1
0.1], 'FaceAlpha',0.25);
Diseno(tam+5)=patch(hp3(:,1)+dx,hp3(:,2)+dy,hp3(:,3)+dz,[0.1 0.1
0.1], 'FaceAlpha',0.25);
Diseno(tam+6)=patch(hp4(:,1)+dx,hp4(:,2)+dy,hp4(:,3)+dz,[0.1 0.1
0.1], 'FaceAlpha',0.25);

% helices

x=1.4142*cos(th)*0.06;
y=1.4142*sin(th)*0.5;
z=0.22*ones(1,length(th));

h1=[x'+1.7 y'+1 z']*scale*R;
h2=[x'+1.7 y'+1 z']*scale*R;
h3=[x'-1.7 y'+1 z']*scale*R;
h4=[x'-1.7 y'+1 z']*scale*R;

h5=[x'+1.7 y'-1 z']*scale*R;
h6=[x'+1.7 y'-1 z']*scale*R;
h7=[x'-1.7 y'-1 z']*scale*R;
h8=[x'-1.7 y'-1 z']*scale*R;

Diseno(tam+9)=patch(h1(:,1)+dx,h1(:,2)+dy,h1(:,3)+dz,[0.1 0.1 0.1]);
Diseno(tam+10)=patch(h2(:,1)+dx,h2(:,2)+dy,h2(:,3)+dz,[0.1 0.1 0.1]);
Diseno(tam+11)=patch(h3(:,1)+dx,h3(:,2)+dy,h3(:,3)+dz,[0.1 0.1 0.1]);
Diseno(tam+12)=patch(h4(:,1)+dx,h4(:,2)+dy,h4(:,3)+dz,[0.1 0.1 0.1]);
Diseno(tam+13)=patch(h5(:,1)+dx,h5(:,2)+dy,h5(:,3)+dz,[0.1 0.1 0.1]);
Diseno(tam+14)=patch(h6(:,1)+dx,h6(:,2)+dy,h6(:,3)+dz,[0.1 0.1 0.1]);
Diseno(tam+15)=patch(h7(:,1)+dx,h7(:,2)+dy,h7(:,3)+dz,[0.1 0.1 0.1]);
Diseno(tam+16)=patch(h8(:,1)+dx,h8(:,2)+dy,h8(:,3)+dz,[0.1 0.1 0.1]);

% centro

num_lados=8;
a=360/num_lados*(pi/180);

```

```

th=0:a:2*pi;
x=1.4142*cos(th)*0.3;
y=1.4142*sin(th)*0.4;
z=0.05*ones(1,length(th));
zb=0.3;
centro=[x' y' z'];
centro1=[x' y' z'+zb];

num_lados=3;
a=360/num_lados*(pi/180);
th=-pi/2:a:2*pi-pi/2;
x=1.4142*cos(th)*0.15;
y=1.4142*sin(th)*0.15-0.1;
z=0.36*ones(1,length(th));

punto_control=[x' y' z']*scale*R*10;
Diseno(tam+21)=patch(punto_control(:,1)+dx,punto_control(:,2)+dy,punto_control(:,3)+
dz,colorp);

for k=1:length(centro(:,1))-1
    sp1=[centro(k,1) centro(k,2) centro(k,3);centro(k+1,1) centro(k+1,2)
centro(k,3); centro(k+1,1) centro(k+1,2) centro1(k,3); centro(k,1) centro(k,2)
centro1(k,3)]*scale*R;
    Diseno(tam+21+k)=patch(sp1(:,1)+dx,sp1(:,2)+dy,sp1(:,3)+dz,[0.1 0.1 0.1]);
    if k==length(centro(:,1))-1
        sp1=[centro(length(centro),1) centro(length(centro),2) centro(k,3); centro(1,1)
centro(1,2) centro(k,3); centro(1,1) centro(1,2)
centro1(k,3);centro(length(centro),1) centro(length(centro),2)
centro1(k,3)]*scale*R;
        Diseno(tam+21+k+1)=patch(sp1(:,1)+dx,sp1(:,2)+dy,sp1(:,3)+dz,[0.1 0.1 0.1]);
    end
end
tam=tam+21+k+1;
centro=centro*scale*R;
centro1=centro1*scale*R;
Diseno(tam+1)=patch(centro(:,1)+dx,centro(:,2)+dy,centro(:,3)+dz,[1 1 1]);
Diseno(tam+2)=patch(centro1(:,1)+dx,centro1(:,2)+dy,centro1(:,3)+dz,[1 1 1]);
tam=tam+2;

% motores
a=pi/10;
th=pi/4:a:2*pi+pi/4;

x=1.4142*cos(th)*0.15;
y=1.4142*sin(th)*0.15;
z=0*ones(1,length(th));

hp1=[x'+1.7 y'+1 z'];
hp2=[x'-1.7 y'+1 z'];

```

```

hp3=[x'-1.7 y'-1 z'];
hp4=[x'+1.7 y'-1 z'];

zb=0.21;

for k=1:length(hp1)-1
    sp1=[hp1(k,1) hp1(k,2) 0;hp1(k+1,1) hp1(k+1,2) 0; hp1(k+1,1) hp1(k+1,2) zb;
hp1(k,1) hp1(k,2) zb]*scale*R;
    Diseno(tam+k)=patch(sp1(:,1)+dx,sp1(:,2)+dy,sp1(:,3)+dz,'y');
if k==length(hp1)-1
    sp1=[hp1(length(hp1),1) hp1(length(hp1),2) 0; hp1(1,1) hp1(1,2) 0; hp1(1,1)
hp1(1,2) zb;hp1(length(hp1),1) hp1(length(hp1),2) zb]*scale*R;
    Diseno(tam+k+1)=patch(sp1(:,1)+dx,sp1(:,2)+dy,sp1(:,3)+dz,'y');
end
end
tam=tam+k+1;
for k=1:length(hp2)-1
    sp1=[hp2(k,1) hp2(k,2) 0;hp2(k+1,1) hp2(k+1,2) 0; hp2(k+1,1) hp2(k+1,2) zb;
hp2(k,1) hp2(k,2) zb]*scale*R;
    Diseno(tam+k)=patch(sp1(:,1)+dx,sp1(:,2)+dy,sp1(:,3)+dz,'y');
if k==length(hp2)-1
    sp1=[hp2(length(hp2),1) hp2(length(hp2),2) 0; hp2(1,1) hp2(1,2) 0; hp2(1,1)
hp2(1,2) zb;hp2(length(hp2),1) hp2(length(hp2),2) zb]*scale*R;
    Diseno(tam+k+1)=patch(sp1(:,1)+dx,sp1(:,2)+dy,sp1(:,3)+dz,'y');
end
end
tam=tam+k+1;
for k=1:length(hp3)-1
    sp1=[hp3(k,1) hp3(k,2) 0;hp3(k+1,1) hp3(k+1,2) 0; hp3(k+1,1) hp3(k+1,2) zb;
hp3(k,1) hp3(k,2) zb]*scale*R;
    Diseno(tam+k)=patch(sp1(:,1)+dx,sp1(:,2)+dy,sp1(:,3)+dz,'y');
if k==length(hp3)-1
    sp1=[hp3(length(hp3),1) hp3(length(hp3),2) 0; hp3(1,1) hp3(1,2) 0; hp3(1,1)
hp3(1,2) zb;hp3(length(hp3),1) hp3(length(hp3),2) zb]*scale*R;
    Diseno(tam+k+1)=patch(sp1(:,1)+dx,sp1(:,2)+dy,sp1(:,3)+dz,'y');
end
end
tam=tam+k+1;
for k=1:length(hp4)-1
    sp1=[hp4(k,1) hp4(k,2) 0;hp4(k+1,1) hp4(k+1,2) 0; hp4(k+1,1) hp4(k+1,2) zb;
hp4(k,1) hp4(k,2) zb]*scale*R;
    Diseno(tam+k)=patch(sp1(:,1)+dx,sp1(:,2)+dy,sp1(:,3)+dz,'y');
if k==length(hp4)-1
    sp1=[hp4(length(hp4),1) hp4(length(hp4),2) 0; hp4(1,1) hp4(1,2) 0; hp4(1,1)
hp4(1,2) zb;hp4(length(hp4),1) hp4(length(hp4),2) zb]*scale*R;
    Diseno(tam+k+1)= patch(sp1(:,1)+dx,sp1(:,2)+dy,sp1(:,3)+dz,'y');
end
end
tam=tam+k+1;

```

```
hp1=hp1*scale*R;  
hp2=hp2*scale*R;  
hp3=hp3*scale*R;  
hp4=hp4*scale*R;
```

```
zb=0.21*scale;
```

```
Diseno(tam+1)=patch(hp1(:,1)+dx, hp1(:,2)+dy, hp1(:,3)+dz, 'y');  
Diseno(tam+2)=patch(hp2(:,1)+dx, hp2(:,2)+dy, hp2(:,3)+dz, 'y');  
Diseno(tam+3)=patch(hp3(:,1)+dx, hp3(:,2)+dy, hp3(:,3)+dz, 'y');  
Diseno(tam+4)=patch(hp4(:,1)+dx, hp4(:,2)+dy, hp4(:,3)+dz, 'y');
```

```
Diseno(tam+7)=patch(hp1(:,1)+dx, hp1(:,2)+dy, hp1(:,3)+zb+dz, 'y');  
Diseno(tam+8)=patch(hp2(:,1)+dx, hp2(:,2)+dy, hp2(:,3)+zb+dz, 'y');  
Diseno(tam+9)=patch(hp3(:,1)+dx, hp3(:,2)+dy, hp3(:,3)+zb+dz, 'y');  
Diseno(tam+10)=patch(hp4(:,1)+dx, hp4(:,2)+dy, hp4(:,3)+zb+dz, 'y');
```

```
end
```

Anexo 4: Artículo científico publicado y presentado de manera virtual en “ 2022 5th International Conference on Control, Robotics and Informatics (ICCRI 2022) “, en “ University of Danang “, Danang, Vietnam, llevado a cabo del 2 al 4 de Abril.

Quadcopter Formation Using Backstepping Control and Dynamic Coupling in Master-Slave Configuration

Rubio-Pecasso J. and López-Gutiérrez R.M.

Engineering, Architecture and Design Faculty, Baja California Autonomous University, Mexico

Email: {rubio.johnny, roslopez}@uabc.edu.mx

Arellano-Delgado A.

CONACYT-UABC Engineering, Architecture and Design Faculty, Baja California Autonomous University, Mexico

Email: adrian.arellano@uabc.edu.mx

Cruz-Hernández C.

Electronics and Telecommunications Department Scientific Research and Advanced Studies Center of Ensenada, Mexico

Email: ccruz@cicese.mx

Abstract—In this work, backstepping formation control of quadcopters in master-slave configuration by using a proposed dynamic coupling is presented. By means of extensive numerical simulations, we show the advantages by comparing the system using backstepping control and backstepping control with the proposed dynamic coupling system. Some cases of interest are studied in order to show several benefits when using the proposed coupling scheme. These cases include different desired trajectories for the master quadcopter, such as circular, concentric and lemniscate, where the slave quadcopter receives the desired trajectory from the master quadcopter through a dynamic coupling using a backstepping control. The obtained results in the numerical simulations show that it is possible to achieve quadcopter formation using backstepping control and dynamic coupling in Master-Slave configuration.

Keywords—Quadcopter, Backstepping Control, Dynamic Coupling, formation, Master-Slave Configuration.

I. INTRODUCTION

As technology steps into the ordinary day and day of people, many applications of the robotic field improves and fulfill an important part of our life. In recent years, unfortunately many natural disasters has occur around the globe, and the need to have technology assistant is very important to save more lives as possible. In addition to the above, many companies use UAVs (Unmanned aerial vehicles) to create patrol routes, which goes from medical companies, agriculture companies to preserve and cultivate their product, and most recently delivery companies. A large number of these applications need the interaction of not just one but several interconnected UAVs to solve a specific task. In this regard, designing algorithms that help to create optimal and efficient routes is very important in the area of the UAVs control.

In the last decades, works related to the development of UAV (Unmanned Aerial Vehicle) have been a spotlight of

study, whether in academic, scientific or the private industry [1].

Among the applications where this technology is used are, for example, the exploration of places of difficult access [2], recognition of fields [3], and delivery related companies [4], among many others use with military [5] and medical purposes [6].

In the UAV classification, there are different models and designs of vehicles, among which are quadcopters, which are consider in this work. In Fig. 1 a quadcopter used in many works in the scientific community is presented.



Figure 1. CrazyFly Quadcopter.

A. State of the art

This branch of technology takes advantage on the great contributions that have been studied within the research community. The latest works show great progress in the quadcopter controllers.

One of the techniques used to reduce the number of variables is the modified backstepping, interconnecting two subsystem, one for angular rotation, and one for linear translation [7].

Usually a non-linear control strategy is proposed, in conditions in which the least amount of variables interfere, where the rotation and translation variables are considered zero, it is when the quadcopter is floating and the drag force [8] gives the control.

In control systems, the sliding mode control is a nonlinear method that alters the dynamics of a system through the discontinuous signal, which forces the system to “slide” along a cross section of the normal behavior of the system, therefore it has a variable control strategy. This method helps as different subsystems can be created to improve stability control [9].

The work has been taken to the field of synchronization and formation of topology and networks of quadcopters, one of the most outstanding work is the synchronization through the comparison of attitude dynamics and trajectory dynamics; this allows the system to decouple into inner-loop, corresponding to the dynamics and the error dynamics in the formation [10].

The results obtained in the search for the state of the art, showed that the most of the works are focused in the quadcopter controller, in none of the works it has been used no form of coupling between two or more quadcopters.

Among the opportunities that were found to continue developing this work are:

- Improving the control system adding a Dynamic Coupling.
- Use topologies to create formations between two or more quadcopters.

B. Body and sections of the work

This work is organize as follows. In Section II, some preliminaries concerning backstepping control and the proposed dynamic coupling are presented. In Section III, the used mathematical model of a quadcopter used and the analysis of the implementation of Backstepping control with dynamic coupling are address. In addition, an anti-collision stage is presented in this Section. Section IV shows the obtained simulation results implemented in Matlab software, where different desired trajectories are analyze. Sections V and VI present discussion and conclusion of this work, respectively.

II. PRELIMINARIES

A. Backstepping Control

In control theory, Backstepping is a technique development to stabilize nonlinear dynamic systems. They are built by subsystems that transmit from an irreducible subsystem using another control method [11].

This method is a recursive system, since it needs another controller to stabilize the subsystem, the process ends when all controllers are stabilized.

Backstepping control is a method suitable for “strict-feedback” or also called “lower triangular”. An example of a system with strict feedback it is presented as follows:

$$\dot{\xi}_1 = f_1(\xi_1) + g_1(\xi_1)\xi_2$$

$$\dot{\xi}_2 = f_2(\xi_1, \xi_2) + g_2(\xi_1, \xi_2)\xi_3$$

...

$$\dot{\xi}_r = f_{r-1}(\xi_1, \xi_2 \dots \xi_{r-1}) + g_{r-1}(\xi_1, \xi_2 \dots \xi_{r-1})\xi_r$$

$$\dot{\xi}_r = f_r(\xi_1, \xi_2 \dots \xi_r) + g_r(\xi_1, \xi_2 \dots \xi_r)u \quad (1)$$

Where $\xi_2, \dots, \xi_r \in \mathbb{R}, u \in \mathbb{R}$ is the input and f_i, g_i for $i = 1$, are the known functions. A typical approach to feedback linearization is that, in most cases, there is cancellation of useful non-linearities. The design of backstepping controller exhibits greater flexibility compared to linearization, since it does not require that the result of dynamic input/output be linear.

Cancellation of non-linearities can be avoid, resulting in less complex controllers. The main idea is to use some of the state variables as “virtual controller”, and depending on the dynamics of each state, design intermediate control laws.

The backsteppnig control design is a recursive procedure where a “Lyapunov function” is derived for the entire system. The recursive procedure can be easily expanded from the nominal case of an integrator-augmented system. This is also known as “Integrator Backstepping” [11].

B. Dynamic Coupling

Dynamic systems are systems in which the state variables follow a series of temporal rules. Generally called systems, since they are represented by a set of equations; they are dynamic because their parameters vary with a time variable.

The dynamic coupling inspired by the Huygens experiment is described in [13] as follows:

$$\dot{X}_m = F(X_m) \quad (2)$$

$$\dot{X}_s = F(X_s) - B_1 h \quad (3)$$

$$\dot{h} = G h - c B_2 (X_m - X_s) \quad (4)$$

Where $X_m, X_s \in \mathbb{R}^n$, are the state vectors of the master and slave system respectively, $h = (h_1, h_2)^T$ with $h_i \in \mathbb{R}, i = 1, 2$, are the state variables of the dynamic coupling, it is assumed that the F vector has smooth behavior, and c is the coupling strength, $B_1 \in \mathbb{R}^{n \times 2}$ and $B_2 \in \mathbb{R}^{2 \times n}$ are the coupling matrix. The B_1 matrix indicates the equations of the slave system applied to the dynamic coupling, and the B_2 matrix indicates the master system states that are coupled in the slave system. The G matrix is defined by:

$$G = \begin{bmatrix} -\alpha & 1 \\ -\gamma_1 & -\gamma_2 \end{bmatrix} \quad (5)$$

Where γ_1, γ_2 and α are positive design constants [14]. Figure 2 a) and b) shows the general layout where we can see the difference of using the proposed dynamic coupling in backstepping control.

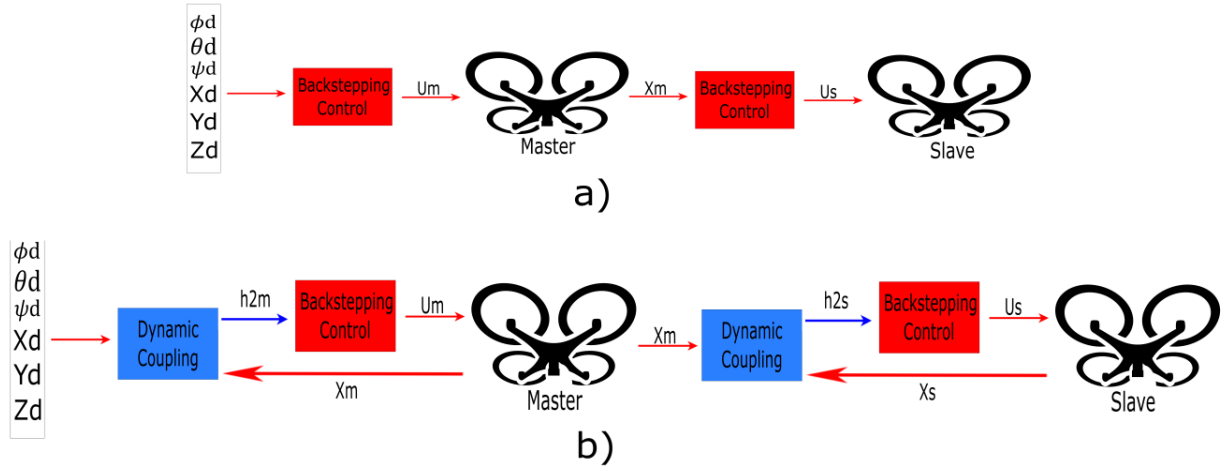


Figure 2. General layout of the proposed dynamic coupling scheme: a) Backstepping control without the proposed dynamic coupling, b) Backstepping control with the proposed dynamic coupling.

III. QUADCOPTER MODEL AND IMPLEMENTED CONTROLLER

A. Mathematical Model of Quadcopter

UAV have six degrees of freedom, those used in space (X, Y, Z) and three angles of inclination in their gravity center (ϕ, θ, ψ) .

The model used in this work is derived from a Newton-Euler Formalism (6).

$$\begin{bmatrix} mI_{3 \times 3} & 0 \\ 0 & I \end{bmatrix} \begin{bmatrix} \dot{V} \\ \dot{\omega} \end{bmatrix} + \begin{bmatrix} \omega \times mV \\ \omega \times I\omega \end{bmatrix} = \begin{bmatrix} F \\ \tau \end{bmatrix} \quad (6)$$

Where, $I \in \mathbb{R}^{3 \times 3}$ is the inertia matrix, V is the velocity vector, and ω is the angular velocity. In Table I more involved variables are listed. The model of the quadcopter can be described as follows:

$$\dot{\zeta} = v \quad (7)$$

$$m\dot{v} = -ge_3 + R_{e3} \left(\frac{b}{m} \right) \Sigma \Omega_i^2 \quad (8)$$

$$\dot{R} = R\hat{\omega} \quad (9)$$

$$I\dot{\omega} = -\omega \times I\omega - \Sigma J_r (\omega \times e_3) \Omega_i + \tau_a \quad (10)$$

TABLE I. VARIABLE DEFINITIONS

Symbol	Definition
ζ	Vector Position
R	Rotation Matrix
$\hat{\omega}$	Symmetric Segment Matrix
ϕ	Roll Angle
θ	Pitch Angle
ψ	Yaw Angle
Ω	Rotor Velocity
	Body Inertia

$I_{x,y,z}$	Rotor Inertia
J_r	Motor Inertia
J_m	Propel Inertia
J_p	Torque
τ_a	Thrust factor
b	Drag factor
d	lever
l	

B. Implemented Backstepping Control

The torque applied to the body of the vehicle on the axle is described as follows:

$$\tau_a = \begin{pmatrix} lb (\Omega_4^2 - \Omega_2^2) \\ lb (\Omega_3^2 - \Omega_1^2) \\ d (\Omega_2^2 + \Omega_4^2 - \Omega_1^2 - \Omega_3^2) \end{pmatrix} \quad (11)$$

If inertia of the rotor is considered and reversing the transmission with the negligent inertia, then the inertia of propulsion of the pairs of rotors are as follows:

$$J_r = J_p - J_m r \quad (12)$$

The complete model is:

$$\ddot{\phi} = \dot{\theta}\dot{\psi} \left(\frac{l_y - l_z}{l_x} \right) - \frac{J_r}{l_x} \dot{\theta}\Omega + \frac{l}{l_x} U_2 \quad (13)$$

$$\ddot{\theta} = \dot{\phi}\dot{\psi} \left(\frac{l_z - l_x}{l_y} \right) + \frac{J_r}{l_y} \dot{\phi}\Omega + \frac{l}{l_y} U_3 \quad (14)$$

$$\ddot{\psi} = \dot{\phi}\dot{\theta} \left(\frac{l_x - l_y}{l_z} \right) + \frac{1}{l_z} U_4 \quad (15)$$

$$\ddot{z} = -g + (\cos \phi \cos \theta) \frac{1}{m} U_1 \quad (16)$$

$$\ddot{x} = (\cos \phi \sin \theta \cos \psi + \sin \phi \sin \psi) \frac{1}{m} U_1 \quad (17)$$

$$\ddot{y} = (\cos \phi \sin \theta \sin \psi - \sin \phi \cos \psi) \frac{1}{m} U_1 \quad (18)$$

The inputs of the system are:

$$U_1 = b (\Omega_1^2 + \Omega_2^2 + \Omega_3^2 + \Omega_4^2) \quad (19)$$

$$U_2 = b (\Omega_4^2 - \Omega_2^2) \quad (20)$$

$$U_3 = b (\Omega_3^2 - \Omega_1^2) \quad (21)$$

$$U_4 = d (\Omega_2^2 + \Omega_4^2 - \Omega_1^2 - \Omega_3^2) \quad (22)$$

$$\Omega = (\Omega_2 + \Omega_4 - \Omega_1 - \Omega_3) \quad (23)$$

The model can be modify by introducing linearization, creating a vector of states in the following way:

$$x_1 = \phi \quad (24)$$

$$x_2 = \dot{x}_1 = \dot{\phi} \quad (25)$$

$$x_3 = \theta \quad (26)$$

$$x_4 = \dot{x}_3 = \dot{\theta} \quad (27)$$

$$x_5 = \psi \quad (28)$$

$$x_6 = \dot{x}_5 = \dot{\psi} \quad (29)$$

$$x_7 = z \quad (30)$$

$$x_8 = \dot{x}_7 = \dot{z} \quad (31)$$

$$x_9 = x \quad (32)$$

$$x_{10} = \dot{x}_9 = \dot{x} \quad (33)$$

$$x_{11} = y \quad (34)$$

$$x_{12} = \dot{x}_{11} = \dot{y} \quad (35)$$

The differential equations are:

$$f(X, U) = \begin{bmatrix} x_2 \\ x_4 x_6 a_1 + x_4 a_2 \Omega + b_1 U_2 \\ x_4 \\ x_2 x_6 a_3 + x_2 a_4 \Omega + b_2 U_3 \\ x_6 \\ x_4 x_2 a_5 + b_3 U_4 \\ x_8 \\ -g + (\cos x_1 \cos x_3) \frac{1}{m} U_1 \\ x_{10} \\ U_x \frac{1}{m} U_1 \\ x_{12} \\ U_y \frac{1}{m} U_1 \end{bmatrix} \quad (36)$$

Where:

$$a_1 = (I_y - I_z) / I_x \quad (37)$$

$$a_2 = -J_R / I_x \quad (38)$$

$$a_3 = (I_z - I_x) / I_y \quad (39)$$

$$a_4 = J_R / I_y \quad (40)$$

$$a_5 = (I_x - I_y) / I_z \quad (41)$$

$$b_1 = l / I_x \quad (42)$$

$$b_2 = l / I_y \quad (43)$$

$$b_3 = l / I_z \quad (44)$$

$$U_x = (\cos x_1 \sin x_3 \cos x_5 + \sin x_1 \sin x_5) \quad (45)$$

$$U_y = (\cos x_1 \sin x_3 \sin x_5 - \sin x_1 \cos x_5) \quad (46)$$

The variable definitions are shown in Table II.

TABLE II. DEFINITIONS

Symbol	Definition	Value
m	Mass	0.650 Kg
I_x	Inertia X axis	$7.5e - 3 \text{ Kg m}^2$
I_y	Inertia Y axis	$7.5e - 3 \text{ Kg m}^2$
I_z	Inertia Z axis	$1.3e - 3 \text{ Kg m}^2$
b	Thrust Coefficient	$3.13e - 5 \text{ N s}^2$
d	Drag coefficient	$7.5e - 7 \text{ N m s}^2$
J_r	Rotor inertia	$6e - 5 \text{ Kg m}^2$
L	Length	0.23 m

The translation components depend on the rotation angles as shown in fig. 3, the system described by (36) ideally constituted by two subsystems; the angular rotation and the linear translation.

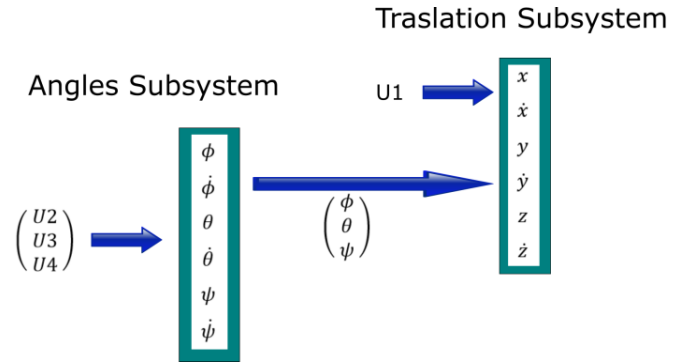


Figure 3. Connection of the subsystems of the overall dynamical system.

The fig.4 describes the scheme advocated for the overall system, divided in a position controller and rotation controller.

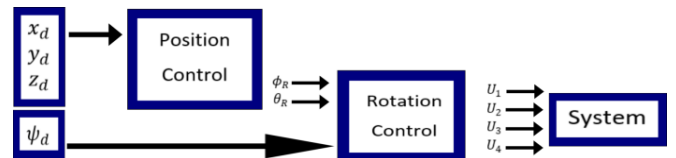


Figure 4. Control Scheme: One set the desired (xd, yd, zd, psi_d), the position controller generates the required (phi_R, theta_R) to the rotation controller.

The first step to implement backstepping control is to design the rotation subsystem, the desired position of the master quadcopter has to be considered. First, the input state for angular rotation is considered, from this derives the position, using the tracking error. The errors are obtained by the following proposed Lyapunov function.

$$V(z_1) = \frac{1}{2} z_1^2 \quad (47)$$

$$\dot{V}(z_1) = z_1(\dot{x}_{1d} - x_2) \quad (48)$$

Introducing a control input for stabilization, we have the following:

$$z_2 = x_2 - \dot{x}_{1d} - a_1 z_1 \quad (49)$$

$$V(z_1, z_2) = \frac{1}{2} (z_1^2 + z_2^2) \quad (50)$$

In the same way, the following can be obtained:

$$z_3 = x_{3d} - x_3 \quad (51)$$

$$z_4 = x_4 - \dot{x}_{3d} - a_3 z_3 \quad (52)$$

$$z_5 = x_{5d} - x_5 \quad (53)$$

$$z_6 = x_6 - \dot{x}_{5d} - a_5 z_5 \quad (54)$$

$$z_7 = x_{7d} - x_7 \quad (55)$$

$$z_8 = x_8 - \dot{x}_{7d} - a_7 z_7 \quad (56)$$

$$z_9 = x_{9d} - x_9 \quad (57)$$

$$z_{10} = x_{10} - \dot{x}_{9d} - a_9 z_9 \quad (58)$$

$$z_{11} = x_{11d} - x_{11} \quad (59)$$

$$z_{12} = x_{12} - \dot{x}_{11d} - a_{11} z_{11} \quad (60)$$

The control input U_2 is extracted from $(\ddot{x}_{1d,2d,3d} = 0)$, satisfying $\dot{V}(z_1, z_2) > 0$.

$$U_2 = \frac{1}{b_1} (z_1 - x_4 x_6 a_1 - x_4 a_2 \Omega - \alpha_1(z_2 + \alpha_1 z_1) - \alpha_2 z_2) \quad (61)$$

The term $\alpha_2 z_2$, when $\alpha_2 > 0$, is added to stabilize z_1 .

$$U_3 = \frac{1}{b_2} (z_3 - x_2 x_6 a_3 - x_2 a_4 \Omega - \alpha_3(z_4 + \alpha_3 z_3) - \alpha_3 z_3) \quad (62)$$

$$U_4 = \frac{1}{b_3} (z_5 - x_2 x_4 a_5 - \alpha_5(z_6 + \alpha_5 z_5) - \alpha_6 z_6) \quad (63)$$

Now designing the translation subsystem, the attitude control U_1 is obtained by using the same approach as before.

$$U_1 = \frac{m}{\cos x_1 \cos x_3} (z_7 + g - \alpha_7(z_8 + \alpha_7 z_7) - \alpha_8 z_8) \quad (64)$$

Movement control in axis (X, Y) is described as follow:

$$U_x = \frac{m}{U_1} (z_9 - \alpha_9(z_{10} + \alpha_9 z_9) - \alpha_{10} z_{10}) \quad (65)$$

$$U_y = \frac{m}{U_1} (z_{11} - \alpha_{11}(z_{12} + \alpha_{11} z_{11}) - \alpha_{12} z_{12}) \quad (66)$$

To perform the calculation of ϕ_R , θ_R , they are obtained from the desired positions as follows:

$$U_z = m(z_7 - \alpha_7(z_8 + \alpha_7 z_7) - \alpha_8 z_8) \quad (67)$$

Now, using the X, Y positions, the following system of equations is obtained:

$$\begin{pmatrix} U_x \\ U_y \end{pmatrix} = (U_z + mg) \begin{bmatrix} \cos \psi_d & \frac{\sin \psi_d}{\cos \theta_d} \\ \sin \psi_d & \frac{-\cos \psi_d}{\cos \theta_d} \end{bmatrix} \begin{bmatrix} \tan(\theta_d) \\ \tan(\phi_d) \end{bmatrix} \quad (68)$$

Finding the solutions of the system, it is as follows:

$$\tan(\theta_R) = \frac{1}{U_z + mg} (\cos \psi_d U_x + \sin \psi_d U_y) \quad (69)$$

$$\tan(\phi_R) = \frac{\cos(\theta_d)}{U_z + mg} (\sin \psi_d - \cos \psi_d U_y) \quad (70)$$

Performing to the required angles:

$$\theta_R = \tan^{-1} \left(\frac{1}{U_z + mg} (\cos \psi_d U_x + \sin \psi_d U_y) \right) \quad (71)$$

$$\phi_R = \tan^{-1} \left(\frac{\cos(\theta_d)}{U_z + mg} (\sin \psi_d - \cos \psi_d U_y) \right) \quad (72)$$

C. Implemented Dynamic Coupling

The next equations represent the Dynamic Coupling used in the Master Quadcopter:

$$h_\phi \begin{cases} \dot{h}_{1\phi} = -h_{1\phi} + h_{2\phi} \\ \dot{h}_{2\phi} = -\gamma_1 h_{1\phi} - \gamma_2 h_{2\phi} - k(\phi_d - x_1) \end{cases} \quad (73)$$

$$h_\theta \begin{cases} \dot{h}_{1\theta} = -h_{1\theta} + h_{2\theta} \\ \dot{h}_{2\theta} = -\gamma_1 h_{1\theta} - \gamma_2 h_{2\theta} - k(\theta_d - x_3) \end{cases} \quad (74)$$

$$h_\psi \begin{cases} \dot{h}_{1\psi} = -h_{1\psi} + h_{2\psi} \\ \dot{h}_{2\psi} = -\gamma_1 h_{1\psi} - \gamma_2 h_{2\psi} - k(\psi_d - x_5) \end{cases} \quad (75)$$

$$h_z \begin{cases} \dot{h}_{1z} = -h_{1z} + h_{2z} \\ \dot{h}_{2z} = -\gamma_1 h_{1z} - \gamma_2 h_{2z} - k(z_d - x_7) \end{cases} \quad (76)$$

$$h_x \begin{cases} \dot{h}_{1x} = -h_{1x} + h_{2x} \\ \dot{h}_{2x} = -\gamma_1 h_{1x} - \gamma_2 h_{2x} - k(x_d - x_9) \end{cases} \quad (77)$$

$$h_y \begin{cases} \dot{h}_{1y} = -h_{1y} + h_{2y} \\ \dot{h}_{2y} = -\gamma_1 h_{1y} - \gamma_2 h_{2y} - k(y_d - x_{11}) \end{cases} \quad (78)$$

Where γ_1 and γ_2 are positive constants, and k is the coupling strength. Using the dynamic coupling the errors for the master quadcopter are as follow:

$$z_1 = -\dot{h}_{2\phi} \quad (79)$$

$$z_2 = x_2 - a_2 z_2 \quad (80)$$

$$z_3 = -\dot{h}_{2\theta} \quad (81)$$

$$z_4 = x_4 - a_3 z_3 \quad (82)$$

$$z_5 = -\dot{h}_{2\psi} \quad (83)$$

$$z_6 = x_6 - a_5 z_5 \quad (84)$$

$$z_7 = -\dot{h}_{2z} \quad (85)$$

$$z_8 = x_8 - a_7 z_7 \quad (86)$$

$$z_9 = \dot{h}_{2x} \quad (87)$$

$$z_{10} = x_{10} - a_9 z_9 \quad (88)$$

$$z_{11} = -\dot{h}_{2y} \quad (89)$$

$$z_{12} = x_{12} - a_{11} z_{11} \quad (90)$$

For the slave system, state equations are:

$$f(X_s, U_s) = \begin{pmatrix} x_{2s} \\ x_{4s} x_{6s} a_{1s} + x_{4s} a_{2s} \Omega + b_1 U_{2s} \\ x_{4s} \\ x_{2s} x_{6s} a_{3s} + x_{2s} a_{4s} \Omega + b_2 U_{3s} \\ x_{6s} \\ x_{4s} x_{2s} a_{5s} + b_3 U_{4s} \\ x_{8s} \\ -g + (\cos x_{1s} \cos x_{3s}) \frac{1}{m} U_{1s} \\ x_{10s} \\ U_{xs} \frac{1}{m} U_{1s} \\ x_{12s} \\ U_{ys} \frac{1}{m} U_{1s} \end{pmatrix} \quad (91)$$

The implementation of the dynamic coupling in the slave system is described through the following:

$$h_{\phi s} \begin{cases} \dot{h}_{1\phi s} = -h_{1\phi s} + h_{2\phi s} \\ \dot{h}_{2\phi s} = -\gamma_1 h_{1\phi s} - \gamma_2 h_{2\phi s} - k(\beta_1 x_1 - x_{1s} - \Delta_\phi) \end{cases} \quad (92)$$

$$h_{\theta s} \begin{cases} \dot{h}_{1\theta s} = -h_{1\theta s} + h_{2\theta s} \\ \dot{h}_{2\theta s} = -\gamma_1 h_{1\theta s} - \gamma_2 h_{2\theta s} - k(\beta_2 x_3 - x_{3s} - \Delta_\theta) \end{cases} \quad (93)$$

$$h_{\psi s} \begin{cases} \dot{h}_{1\psi s} = -h_{1\psi s} + h_{2\psi s} \\ \dot{h}_{2\psi s} = -\gamma_1 h_{1\psi s} - \gamma_2 h_{2\psi s} - k(\beta_3 x_5 - x_{5s} - \Delta_\psi) \end{cases} \quad (94)$$

$$h_{zs} \begin{cases} \dot{h}_{1zs} = -h_{1zs} + h_{2zs} \\ \dot{h}_{2zs} = -\gamma_1 h_{1zs} - \gamma_2 h_{2zs} - k(\beta_4 x_7 - x_{7s} - \Delta_z) \end{cases} \quad (95)$$

$$h_{xs} \begin{cases} \dot{h}_{1xs} = -h_{1xs} + h_{2xs} \\ \dot{h}_{2xs} = -\gamma_1 h_{1xs} - \gamma_2 h_{2xs} - k(\beta_5 x_9 - x_{9s} - \Delta_x) \end{cases} \quad (96)$$

$$h_{ys} \begin{cases} \dot{h}_{1ys} = -h_{1ys} + h_{2ys} \\ \dot{h}_{2ys} = -\gamma_1 h_{1ys} - \gamma_2 h_{2ys} - k(\beta_6 x_{11} - x_{11s} - \Delta_y) \end{cases} \quad (97)$$

Where, $\Delta_x, \Delta_y, \Delta_z$ are displacements of (X, Y, Z) , respectively. These constant displacements are introduced to separate the quadcopters in the (X, Y, Z) . Note that a sign variable β_i , with $i=1, 2, \dots, 6$ is introduced to generate antiphase synchronization. The errors of the slave quadcopter are as follow:

$$z_{1s} = -\dot{h}_{2\phi s} \quad (98)$$

$$z_{2s} = x_{2s} - \alpha_{1s} * z_{1s} \quad (99)$$

$$z_{3s} = -\dot{h}_{2\theta s} \quad (100)$$

$$z_{4s} = x_{4s} - \alpha_{3s} * z_{3s} \quad (101)$$

$$z_{5s} = -\dot{h}_{2\psi s} \quad (102)$$

$$z_{6s} = x_{6s} - \alpha_{5s} * z_{5s} \quad (103)$$

$$z_{7s} = -\dot{h}_{2zs} \quad (104)$$

$$z_{8s} = x_{8s} - \alpha_{7s} * z_{7s} \quad (105)$$

$$z_{9s} = -\dot{h}_{2xs} \quad (106)$$

$$z_{10s} = x_{10s} - \alpha_{9s} * z_{9s} \quad (107)$$

$$z_{11s} = -\dot{h}_{2ys} \quad (108)$$

$$z_{12s} = x_{12s} - \alpha_{11s} * z_{11s} \quad (109)$$

D. Anti-collision Stage

The propose coupling scheme, it does work properly, however, it does not have a system that can avoid crashes between the quadcopters, therefore, it is resorted to modifying the dynamic coupling, based in a similar model [18], adapted for this work. The anti-collision implemented in this work is as follow:

$$\dot{h}_{2z} = -\gamma_1 h_{1z} - \gamma_2 h_{2z} - k(\beta_4 x_7 - x_{7s} - \Delta_z - r_{ms} \text{sign}(x_7 - x_{7s})) \quad (110)$$

$$\dot{h}_{2x} = -\gamma_1 h_{1x} - \gamma_2 h_{2x} - k(\beta_5 x_9 - x_{9s} - \Delta_x - r_{ms} \text{sign}(x_9 - x_{9s})) \quad (111)$$

$$\dot{h}_{2y} = -\gamma_1 h_{1y} - \gamma_2 h_{2y} - k(\beta_6 x_{11} - x_{11s} - \Delta_y - r_{ms} \text{sign}(x_{11} - x_{11s})) \quad (112)$$

Where, r_{ms} is the anti-collision coefficient and m is the collision threshold, which can take the following values:

$$r_{ms} = \begin{cases} 0, & \text{if } |\vec{r}_{ms}| > m, \\ r, & \text{if } |\vec{r}_{ms}| \leq m. \end{cases} \quad (113)$$

In the Euclidean norm formed by the position of each of axis, the magnitude is obtained, that is:

$$\|\vec{L}_{ms}\| = \sqrt{(x_9 - x_{9s})^2 + (x_{11} - x_{11s})^2 + (x_7 - x_{7s})^2} \quad (114)$$

Where, $\text{sign}(L_{ms})$, is the sign function represented as:

$$\text{sign}(L_{ms}) = \begin{cases} 1, & \text{if } L_{ms} > 0, \\ 0, & \text{if } L_{ms} = 0, \\ -1, & \text{if } L_{ms} < 0. \end{cases} \quad (115)$$

IV. SIMULATIONS RESULTS

All simulations and analysis are implemented in Matlab 2019, using ODE 45 function to solve Differential Equations. Different configurations of desire trajectories are shown to demonstrate the correct performance of the controller design for this work. In Fig. 5 the block diagram of the proposed coupling scheme is presented.

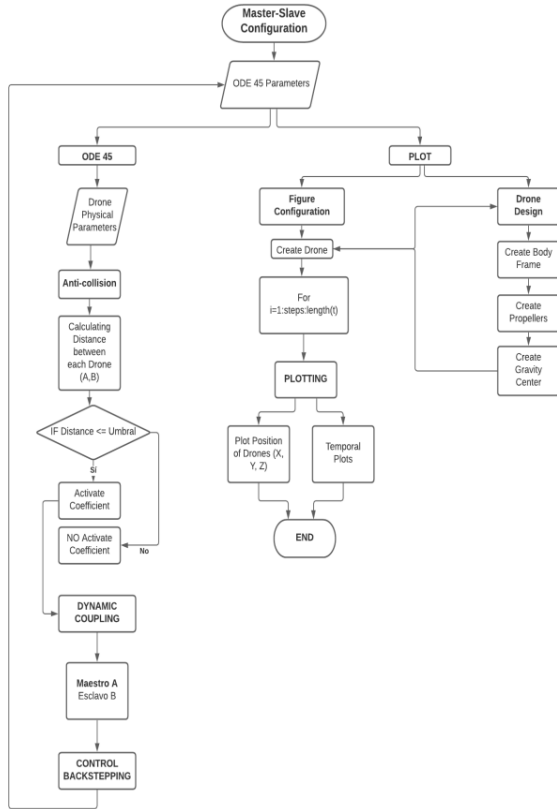


Figure 5. Master-Slave block diagram of the proposed coupling scheme.

A. Single Quadcopter: lemniscate trajectory control

The lemniscate trajectory is represented by the following equations:

$$X_d = 2.5 \cos(0.2 t) \quad (116)$$

$$Y_d = 2.5 \sin(0.4 t) \quad (117)$$

$$Z_d > 0 \quad (118)$$

Where X_d, Y_d, Z_d , represent the desired position on each axis of the plane, and the desire rotation angle is $= 10^\circ$.

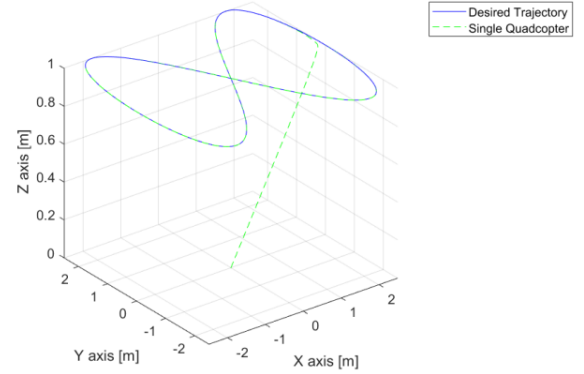


Figure 6. Single Quadcopter: lemniscate Trajectory.

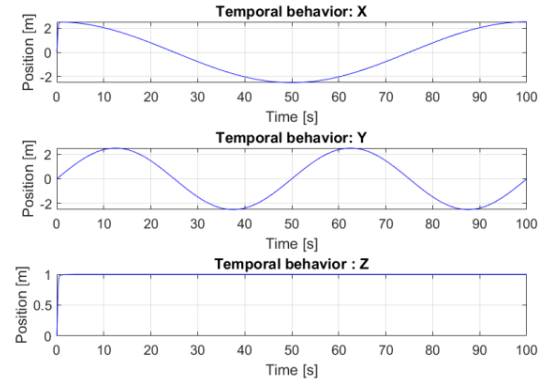


Figure 7. Single Quadcopter: lemniscate Trajectory Temporal.

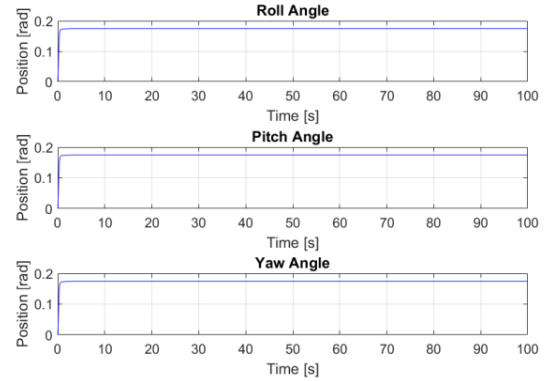


Figure 8. Single Quadcopter: lemniscate Rotation Angles.

Figures 6, 7, and 8 show the simulation plots describing a lemniscate trajectory, temporal behavior of the positions (X,Y,Z), and the stability behavior the rotation angles, respectively, where is demonstrated that the quadcopter tracking desire trajectory is having a good performance.

B. Single Quadcopter: Circle trajectory control

The next equations represents a circle trajectory as follow:

$$Xd = 3 \sin\left(\left(\frac{2\pi}{P}\right) * t\right) \quad (119)$$

$$Yd = 3 \cos\left(\left(\frac{2\pi}{P}\right) * t\right) \quad (120)$$

$$Zd > 0 \quad (121)$$

The desired rotation angle $\psi_d = 10^\circ$.

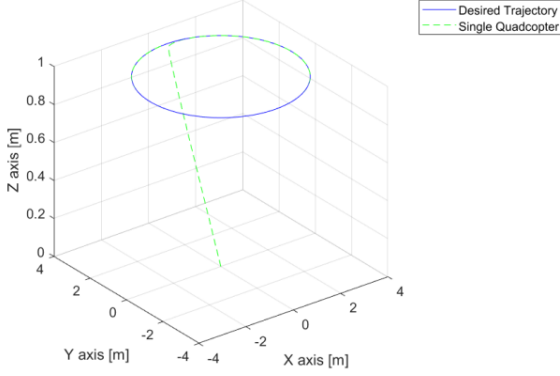


Figure 9. Single Quadcopter: Circle Trajectory.

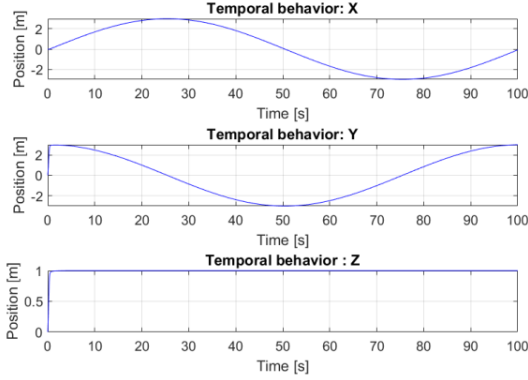


Figure 10. Single Quadcopter: Circle Temporal Trajectory.

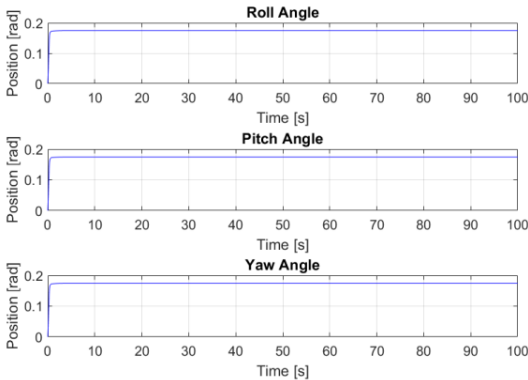


Figure 11. Single Quadcopter: Circle Rotation Angles.

Figures 9, 10, and 11 represent the circular tracking desire trajectory of a single quadcopter, temporal position and rotation angles, respectively, showing a good performance of the system.

C. Master-Slave Configuration: lemniscate Trajectory

The next simulation represents the behavior of the Master-Slave Configuration following a lemniscate trajectory, where the anti-collision is working, with values $m = 1$, $r_{ms} = 0.1$, of the anti-collision coefficient and anti-collision threshold respectively, these same values used in the anti-collision are used in the simulations ahead.

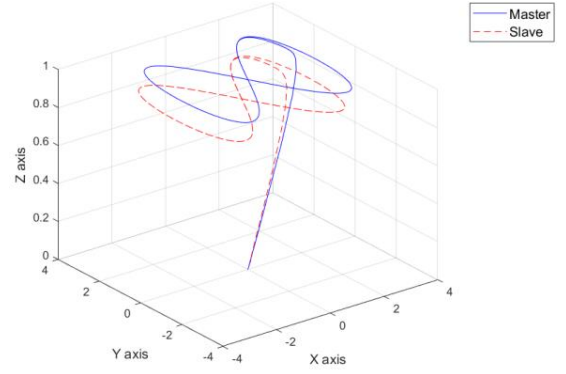


Figure 12. Master-Slave: lemniscate Trajectory.

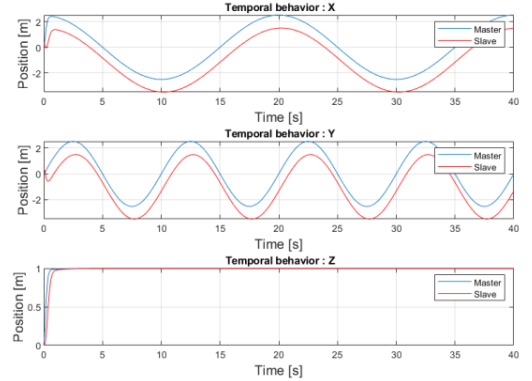


Figure 13. Master-Slave: lemniscate Trajectory Temporal Behavior.

Figures 12 and 13, described the trajectory of a Master-Slave configuration, where the red dotted line represent the slave quadcopter. Note that $\Delta_x = \Delta_y = \Delta_z = 0$, and the slightly move is due to the correct performance of the anti-collision stage.

D. Master-Slave Configuration: Circle Trajectory

Following the Master-Slave Configuration now is applied in a Circular trajectory, in this example is applied antiphase in X and Y axis, with $\beta_5 = -1$ and $\beta_6 = -1$.

Antiphase stage showed in Fig. 14 and 15, where the red dotted line represent the Slave quadcopter that is set in antiphase with respect of the Master.

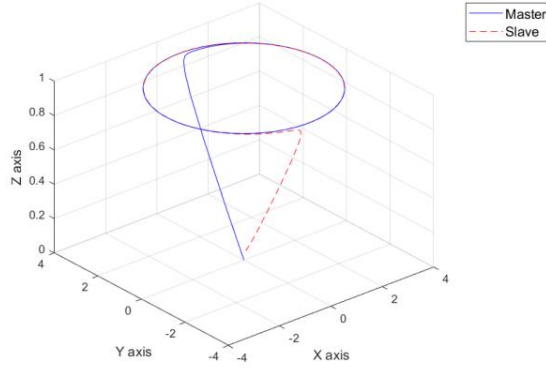


Figure 14. Master-Slave: Circle Trajectory.

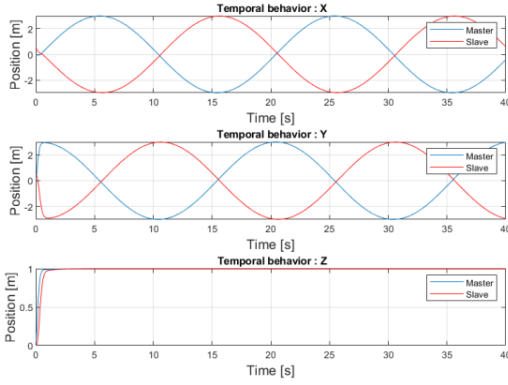


Figure 15. Master-Slave: Circle Trajectory Temporal Behavior.

E. Master-Slave Concentric Configuration: Lemniscate Trajectory

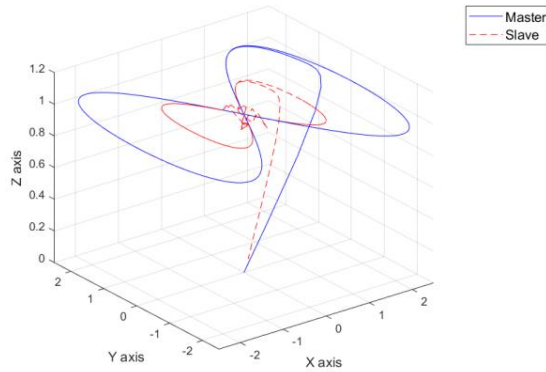


Figure 16. Master-Slave Concentric: Lemniscate Trajectory.

This simulation represents the behavior of the Master-Slave Concentric Configuration following an lemniscate trajectory, and the displacement between master and slave $\Delta_x = 0.5 * x_9$ and $\Delta_y = 0.5 * x_{11}$, these values are used in the simulations ahead.

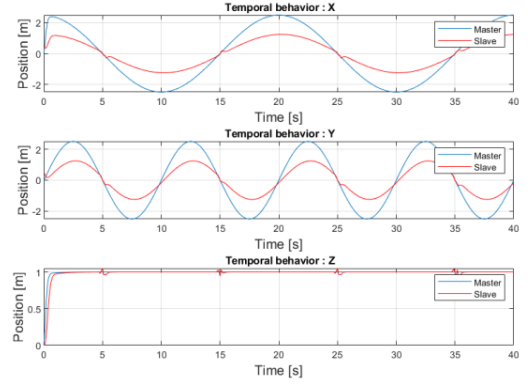


Figure 17. Master-Slave Concentric: Lemniscate Trajectory Temporal Behavior.

F. Master-Slave Concentric Configuration: Circle Trajectory

Now representing the behavior of the Master-Slave Concentric Configuration with a circle trajectory, we have Figs. 18 and 19 that show a circular concentric trajectory from the slave quadcopter, in this particular example the anti-collision stage is not compromised due to the distance between quadcopters.

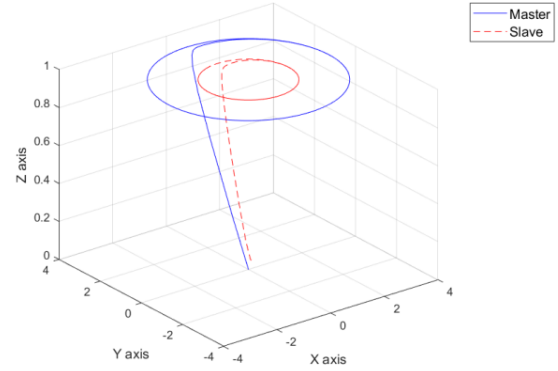


Figure 18. Master-Slave Concentric: Circle Trajectory.

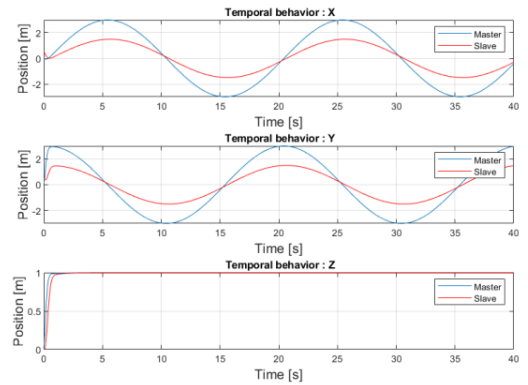


Figure 19. Master-Slave Concentric: Circle Trajectory Temporal Behavior.

V. DISCUSSIONS

In this work, we proposed using a backstepping control and a dynamic system as coupling. Below is a comparative analysis between two cases, using and not using a dynamic coupling in a backstepping control.

A. Comparison with respect of computability time

For the experiments different gains ($\alpha_1 \dots \alpha_{12}$) and coupled constant (k) were used in both systems, writing two different programs in Matlab and using the function tic toc, to know the time elapsed in each system.

TABLE III. COMPARISON 1

	Backstepping Control	Backstepping Control with Dynamic Coupling
Gains	$\alpha_1 \dots \alpha_{12} = 200$	$\alpha_1 \dots \alpha_{12} = 200$ $k = 200$
Computational time	3.125871 sec.	1.287702 sec.

With the results obtained in Table III, where there are equal gains in both systems, it can be concluded that the computation time is 250% faster using a dynamic system.

To analyze how both systems behave with small or minimal gains, the system was analyzed with the following parameters.

TABLE IV. COMPARISON 2

	Backstepping Control	Backstepping Control with Dynamic Coupling
Gains	$\alpha_1 \dots \alpha_{12} = 1$	$\alpha_1 \dots \alpha_{12} = 1$ $k = 1$
Computational time	0.582050 sec.	0.355879 sec.

As is demonstrated in Table IV, with small gains, it was observed that the behavior transient response in the dynamic system was more affected than in the backstepping control; in terms of computation time, the one using dynamic system is still better, but now with around 75% compared to backstepping control.

TABLE V. COMPARISON 3

	Backstepping Control	Backstepping Control with Dynamic Coupling
Gains	$\alpha_1 \dots \alpha_{12} = 2000$	$\alpha_1 \dots \alpha_{12} = 2000$ $k = 2000$
Computational time	23.215402 sec.	8.319175 sec.

In Table V, we can see that the computational time is much bigger in the backstepping control without dynamic coupling, than other examples, which demonstrates an exponential behavior in time with respect of the gains.

B. Comparison with respect of the behavior of the inputs

Regarding the complexity time in which both systems are solved, has similar behavior; however, the greatest advantage of using dynamic coupling are the magnitudes in the system inputs (U_1, U_2, U_3, U_4), in which the dynamic coupling system presented smaller magnitude inputs, which translates into less pronounced peaks; this last part is important when the system is implemented with physical equipment.

C. Advantages of using Backstepping Control with Dynamic Coupling

In this work is concluded that using Dynamic Coupling is a better option than using just Backstepping Control, although the simulations between both systems are similar, there are the following advantages that reinforce the idea of this work.

1. Intermediary System, every computation and calculation goes first by Dynamic Coupling, shown in Fig. 3.
2. The magnitudes of the inputs are smaller after passing through the Dynamic Coupling.
3. The peaks that are caused by the inputs are filtered by the dynamic system, which protect the physical equipment in an experimental implementation.

VI. CONCLUSION

Both configurations works in the implementation, but using backstepping control with dynamic coupling has advantages in computational time, input behavior and the most important advantage the implementation of an intermediary system via Dynamic Coupling. Is shown that the work presented showed the expected results, using backstepping control with dynamic coupling that be implemented in a Master-Slave configuration with different trajectories and applications.

CONFLICT OF INTEREST

The authors declare no conflict of interest.

AUTHOR CONTRIBUTION

Rubio-Pecasso J. conducted, design, analyzed and wrote the content of this paper, **López-Gutiérrez R.M.** conducted and supervised the work, **Arellano-Delgado A.** conducted, analyzed and corroborate the research shown in this paper, and **Cruz-Hernández C.** supervised the work; all authors had approved the final version.

ACKNOWLEDGEMENTS

This work was supported by the Ensenada Center for Scientific Research and Higher Education (CICESE) and the Mexican Council for Science and Technology (CONACYT) through the Research Project on Basic Science, ref. 166654 (A1-S-31628).

REFERENCES

- [1] A. McKenna. (2013). The future of Drone use: Opportunities and Threats from Ethical and Legal Perspectives,. United Kingdom: Asser Press - Springer, p. 355.
- [2] O.E.U. (2015). Community Research and Development Information Service, Retrieved October 9 2020.
- [3] Floreano D. (2015). Science, technology and the future of small autonomous drones. *Nature*. 521 (7553): 460–466. Bibcode:2015Natur.521..460F. doi:10.1038/nature14542. PMID 26 017445. S2CID 4463263.
- [4] Carlson, Daniel F., (2018). Adapting open-source drone autopilots for real-time iceberg observations. *MethodsX*, p. 1059-1072.
- [5] Ackerman, Spencer, Shachtman, Noah. (2012). Almost 1 in 3 U.S. Warplanes Is a Robot. Retrieved from WIRED, 9 October 2020.
- [6] Peck A., (2020). Coronavirus Spurs Perceptor's Drone-in-a-box Surveillance Solution. *DRONELIFE*, retrieved 9 October 2020.
- [7] H. Bouadi, M. Bouchoucha, and M. Tadjine. Sliding Mode Control based on Backstepping Approach for an UAV Type-Quadrotor. *World Academy of Science, Engineering and Technology International Journal of Mechanical, Aerospace, Industrial, Mechatronic and Manufacturing Engineering* Vol:1, No:2, 2007.
- [8] Ashfaq Ahmad Mian, Wang Daobo. Modeling and Backstepping-based Nonlinear Control Strategy for a 6 DOF Quadrotor Helicopter. *Chinese Journal of Aeronautics*, 21, (2008) 261-268.
- [9] Sahul M.P.V., V. Naveen Chandler, Thomas Kurian. (2014) A novel method on Disturbance Rejection PID Controller for Quadcopter based on Optimization algorithm. Third international Conference on Advances in control and optimization of Dynamical Systems.
- [10] EDO USM (2019). Aplicación del metodo Lyapunov en un ejemplo. Recuperado el 27 de Octubre del 2020, de: https://www.youtube.com/watch?v=D-tnLvqr1_0
- [11] Kotkovic P. (1992). The joy of feedback: nonlinear and adaptive. *IEEE control magazine*. 12(3): 7-17.
- [12] S.N. (2022). CrazyFly, Silicio.mx, retrieved from: <https://silicio.mx/crazyfly-20>, January 8, 2022.
- [13] York Vara Herrera. (Master Degree Tesis 2021). Diseño de Trayectorias para Robots Móviles Tipo Diferencial. Universidad Autónoma de Baja California.
- [14] Pena Ramirez, J., Arellano-Delgado, A., y Nijmeijer, H. (2018). Enhancing master-slave synchronization: The effect of using a dynamic coupling. *Physical Review E*, 98(1): 1–10.
- [15] López Parra A. (Master Degree Tesis 2017). Formación en grupos de robots móviles. Centro de Investigación Científica y de Educación Superior de Ensenada.
- [16] López-Parra A., Arellano-Delgado A., López-Gutiérrez R.M., Cruz-Hernández C., Formation in networks in mobile robots. *Proceedings of the Eighth International Conference on Complex Systems Design & Management, CSD&M Paris 2017*.
- [17] A. López-Parra, A. Arellano-Delgado, C. Cruz-Hernández, R. Martínez-Clark, J.J. Cetina-Denis. Formación en grupos de robots móviles. *Latin American Conference on Automatic Control*, 13 a 15 de octubre de 2016, Medellín, Colombia.
- [18] Erik Hernández Valente. (Master Degree Tesis 2018). Sincronización de robots terrestres y aéreos. Universidad Autónoma de Baja California.
- [19] Wang X.F. Complex networks: Topology, dynamics and synchronization. *International Journal Bifurcation and Chaos*, 12(5), (2002) 885-916.
- [20] Khalil H. (1996) *Nonlinear systems*. Prentice Hall.
- [21] Fradkov A., Miroshnik I.V. y Nikiforov O. (1999) *Nonlinear and adaptive control of complex systems*. Kluwer Academic Publishers.
- [22] Martínez C. (2014). Diseño de un controlador para la formación de grupos de robots móviles. Universidad Autónoma de Baja California

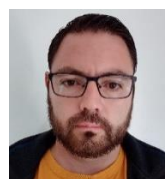
AUTHORS BACKGROUND



Rubio-Pecasso J. has a bachelor's degree in Mechatronics Engineering from Center of Technical and Higher Studies (CETYS) in Mexico. His work experience goes from the Mechanical and Robotics Design to design of algorithms in the Computer Science and AI. Field. Currently he is pursuing his Master's Degree in Control and Robotics at the Engineering, Architecture and Design Faculty from Baja California Autonomous University (UABC) in Mexico.



López-Gutiérrez R. M. was born on 10-11-1972. She is a Professor of Electronics Engineering in Baja California Autonomous University since 2001. She received her Master Science degree and Ph.D. degree in Electronics and Telecommunications from CICESE, Mexico in 1996 and 2003, respectively. Her research interests involve synchronization of complex systems and applications.



Arellano-Delgado A. Doctor of Science in electrical from the Baja California Autonomous University (UABC), Ensenada, B.C., Mexico in 2013. Since 2008, he has been working at the UABC as a professor of different subjects related to electronics. He did a postdoctoral stay at the Electronics and Telecommunications Department Scientific Research and Advanced Studies Center of Ensenada (CICESE) in 2014-2016. He is currently a CONACYT research fellow commissioned to the UABC (Project no. 3059). His topics of interest include synchronization and control of dynamic systems, complex systems, chaotic systems, cryptography, and robotics.



Cruz-Hernández C. received the M.S. and Ph.D. degrees in electrical engineering from CINVESTAV, México, in 1991 and 1995, respectively. Since 1995, he is with the Department of Electronics and Telecommunications of the Scientific Research and Advanced Studies of Ensenada (CICESE), where, he is current Professor of Automatic Control. His research interests include multimode oscillations of coupled oscillators, nonlinear systems analysis, and synchronization and control of complex dynamical system.